



Calhoun: The NPS Institutional Archive
DSpace Repository

Theses and Dissertations

1. Thesis and Dissertation Collection, all items

1996-12

Minimum vibration maneuvers using input
shaping and pulse-width, pulse frequency
modulated thruster control

Buck, Nicholas V.

Monterey, California. Naval Postgraduate School

<http://hdl.handle.net/10945/31955>

Downloaded from NPS Archive: Calhoun



Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community. Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

<http://www.nps.edu/library>

NAVAL POSTGRADUATE SCHOOL

Monterey, California



19970527 030

THESIS

**MINIMUM VIBRATION MANEUVERS USING
INPUT SHAPING AND PULSE-WIDTH, PULSE-
FREQUENCY MODULATED THRUSTER
CONTROL**

by

Nicholas V. Buck

December 1996

Thesis Advisor:

Brij Agrawal

Approved for public release; distribution is unlimited.

DTIC QUALITY INSPECTED 1

| REPORT DOCUMENTATION PAGE | | | Form Approved OMB No. 0704-0188 | |
|--|--|---|----------------------------------|--|
| Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503. | | | | |
| 1. AGENCY USE ONLY (Leave blank) | 2. REPORT DATE December 1996. | 3. REPORT TYPE AND DATES COVERED Engineer's Thesis | | |
| 4. TITLE AND SUBTITLE Minimum Vibration Maneuvers Using Input Shaping and Pulse-Width, Pulse-Frequency Modulated Thruster Control | | 5. FUNDING NUMBERS | | |
| 6. AUTHOR(S) Nicholas V. Buck | | | | |
| 7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey CA 93943-5000 | | 8. PERFORMING ORGANIZATION REPORT NUMBER | | |
| 9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) | | 10. SPONSORING/MONITORING AGENCY REPORT NUMBER | | |
| 11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. | | | | |
| 12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited. | | 12b. DISTRIBUTION CODE | | |
| 13. ABSTRACT (maximum 200 words) Minimizing the modal vibration induced by on-off thrusters is a challenging problem for designers of flexible spacecraft. This thesis presents the first study of Pulse-Width, Pulse-Frequency (PWPF) modulated thruster control using the method of command input shaping. Input shaping for systems with linear actuators has been successfully developed to reduce modal vibrations. Recently, this method has been extended to systems with on-off actuators to some degree. However, existing approaches require complicated non-linear optimization and result in bang-bang control action. Bang-bang thruster operation on flexible spacecraft is propellant-intensive and causes frequent thruster switches. In this thesis, a new approach integrating command input shaping with PWPF-modulated thruster control is developed to minimize residual vibration in maneuvers and to reduce propellant consumption. To realize this approach, an in-depth analysis of the PWPF modulator is first conducted to recommend parameter settings. Next, command input shapers are designed and integrated with the PWPF modulator. Simulation verifies the efficacy of this technique in reducing modal vibration. Lastly, robustness analyses are performed and demonstrate the method's insensitivity to frequency and damping uncertainty. | | | | |
| 14. SUBJECT TERMS Input Shaping, Pulse-Width Pulse-Frequency, Flexible Spacecraft Slewing, Thruster Control. | | 15. NUMBER OF PAGES 162 | | |
| | | 16. PRICE CODE | | |
| 17. SECURITY CLASSIFICATION OF REPORT Unclassified | 18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified | 19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified | 20. LIMITATION OF ABSTRACT UL | |

Approved for public release; distribution is unlimited.

**MINIMUM VIBRATION MANEUVERS USING INPUT SHAPING AND PULSE-
WIDTH, PULSE FREQUENCY MODULATED THRUSTER CONTROL**

Nicholas V. Buck

Lieutenant Commander, United States Navy

B.S.E.E., United States Naval Academy, 1984

Submitted in partial fulfillment
of the requirements for the degree of

AERONAUTICAL AND ASTRONAUTICAL ENGINEER

from the


NAVAL POSTGRADUATE SCHOOL

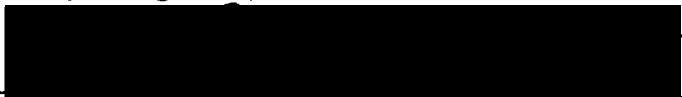
December 1996


Author:


Nicholas V. Buck

Approved by:


Brij N. Agrawal, Thesis Advisor


Gangbing Song, Second Reader


Daniel J. Collins, Chairman
Department of Aeronautics and Astronautics

ABSTRACT

Minimizing the modal vibration induced by on-off thrusters is a challenging problem for designers of flexible spacecraft. This thesis presents the first study of Pulse-Width, Pulse-Frequency (PWPF) modulated thruster control using the method of command input shaping. Input shaping for systems with linear actuators has been successfully developed to reduce modal vibrations. Recently, this method has been extended to systems with on-off actuators to some degree. However, existing approaches require complicated non-linear optimization and result in bang-bang control action. Bang-bang thruster operation on flexible spacecraft is propellant-intensive and causes frequent thruster switches. In this thesis, a new approach integrating command input shaping with PWPF-modulated thruster control is developed to minimize residual vibration in maneuvers and to reduce propellant consumption. To realize this approach, an in-depth analysis of the PWPF modulator is first conducted to recommend parameter settings. Next, command input shapers are designed and integrated with the PWPF modulator. Simulation verifies the efficacy of this technique in reducing modal vibration. Lastly, robustness analyses are performed and demonstrate the method's insensitivity to frequency and damping uncertainty.

TABLE OF CONTENTS

| | |
|--|----|
| I. INTRODUCTION | 1 |
| A. BACKGROUND | 1 |
| B. MOTIVATION FOR RESEARCH..... | 1 |
| 1. Active versus Passive Control | 1 |
| 2. Naval Postgraduate School Research Efforts..... | 2 |
| 3. Potential Impact of Input Shaping Techniques | 2 |
| C. SCOPE OF THESIS..... | 3 |
| II. FLEXIBLE SPACECRAFT SIMULATOR (FSS) SYSTEM MODEL | 5 |
| A. EXPERIMENTAL FACILITY | 5 |
| B. FSS FLEXIBLE APPENDAGE | 6 |
| 1. Description..... | 6 |
| 2. Finite Element Model | 7 |
| C. EQUATIONS OF MOTION..... | 8 |
| 1. Rigid-Elastic Coupling Effect..... | 9 |
| 2. Addition of Piezoceramics to the FSS Model..... | 10 |
| 3. Model Configuration..... | 12 |
| D. STATE SPACE FORMULATION..... | 13 |
| III. PULSE WIDTH, PULSE FREQUENCY MODULATORS | 15 |
| A. BACKGROUND | 15 |
| B. DESCRIPTION OF PWPF MODULATOR..... | 15 |
| 1. Motivation for PWPF Modulated Slewing | 16 |
| 2. Previous Research into PWPF Modulation..... | 17 |
| C. PWPF TIME DOMAIN EQUATIONS | 17 |
| 1. Modulator Time Response..... | 18 |
| 2. System Operating Equations..... | 19 |
| D. PWPF MODULATOR DESIGN ANALYSIS | 23 |
| 1. Static Analysis | 24 |
| 2. Dynamic Analysis..... | 36 |
| 3. Impact of Modulator Parameters on FSS Slewing Performance | 39 |
| 4. Design Recommendations | 45 |
| IV. INPUT SHAPING TECHNIQUES | 47 |
| A. BACKGROUND & DESCRIPTION | 47 |
| B. ADVANTAGES OF INPUT SHAPING | 49 |
| C. DESCRIPTION OF INPUT SHAPERS | 50 |
| 1. Classification by Actuator..... | 50 |
| 2. Classification by Impulse Train..... | 50 |
| D. INPUT SHAPER DESIGN..... | 50 |
| 1. Vibration and Robustness Constraint Equations..... | 51 |
| 2. Constant Amplitude Pulse (CAP) Shaper Constraints..... | 55 |
| 3. Extension to Multiple Modes..... | 57 |

| | |
|---|-----|
| E. SUMMARY | 57 |
| V. SHAPED SLEW MANEUVERS USING LINEAR ACTUATORS | 59 |
| A. APPROACH | 59 |
| B. FSS WITH SINGLE FLEXIBLE MODE | 59 |
| 1. Generation of Shaper Commands | 59 |
| 2. Open Loop System Response | 62 |
| 3. Closed Loop Slew Maneuvers | 64 |
| C. FSS WITH MULTIPLE FLEXIBLE MODES | 66 |
| 1. Target Mode Selection | 66 |
| 2. Generation of Shaper Commands | 67 |
| 3. Open Loop System Response | 71 |
| 4. Closed Loop Slew Maneuvers | 72 |
| D. DISCUSSION | 72 |
| VI. SHAPED SLEW MANEUVERS USING PWPF MODULATOR | 75 |
| A. MOTIVATION | 75 |
| B. INTRODUCTION TO SIMULATIONS | 76 |
| 1. Shaper Selection | 76 |
| 2. Simulation Models | 76 |
| C. FSS WITH SINGLE FLEXIBLE MODE | 77 |
| 1. Open Loop System Response | 78 |
| 2. Closed Loop Slew Maneuvers | 82 |
| 3. Shaper Selection | 83 |
| D. FSS WITH MULTIPLE FLEXIBLE MODES | 83 |
| 1. Description of Simulations | 83 |
| 2. System Response to Slew Maneuvers | 84 |
| 3. Robustness | 87 |
| E. SUMMARY OF RESULTS | 92 |
| VII. CONCLUSIONS AND RECOMMENDATIONS | 93 |
| A. CONCLUSIONS | 93 |
| B. RECOMMENDATIONS FOR FUTURE RESEARCH | 94 |
| 1. Remaining Issues in Input Shaping | 94 |
| 2. Integrated PWPF Modulated Shaper with Active Vibration Control | 94 |
| 3. Applications for Advanced Control Theory | 95 |
| 4. Experimental Validation | 95 |
| APPENDIX A | 97 |
| APPENDIX B | 99 |
| LIST OF REFERENCES | 149 |
| INITIAL DISTRIBUTION LIST | 151 |

ACKNOWLEDGEMENTS

The author wishes to gratefully acknowledge the individuals who have made this thesis possible. Dr. Gangbing Song, who spent countless hours researching the topics of PWPF Modulation and input shaping side by side with the author, has been an inspiration and has passed on the art of effective research. Thanks also go to Dr. Brij Agrawal for having the patience and wisdom to let his students discover their true calling in thesis research. He has made this time an excellent and effective learning process.

Without the help and support of family and friends, this thesis would never have come to fruition. Their patience and guidance has helped the author keep perspective on the important things in life. Thanks go to Hazel, Samuel and Meredith for putting up with all the late nights spent at the laboratory. Their support and unconditional love has given this author strength even in the darkest times. Finally, the author wishes to express gratitude to the crew of the *Medora* for keeping the skipper out of trouble and off the rocks. To Garth Hobson, Jim Craig, and Shawn Lobree, fair winds and following seas.

I. INTRODUCTION

A. BACKGROUND

State-of-the-art space-based systems employing vibration-sensitive payloads demand stringent spacecraft attitude control requirements. Concurrently, launch costs have spurred a drive to optimize payload mass fraction by reducing spacecraft structural mass. However, mass reductions can increase the influence of the structure's flexible modes and complicate pointing performance. As structures get lighter and larger, the modal frequencies encroach on controller bandwidths, making control-structure interaction a major stability issue. Without some sort of compensation, fine attitude control in the presence of flexibility is doubtful.

Many of these systems use thrusters to accomplish station-keeping (translational) maneuvers. Attitude control during such maneuvers must be performed with thrusters since momentum wheels can become saturated due to the required control torques. On the other hand, flexible spacecraft which use on-off (bang-bang) thrusters are subject to modal vibrations which can exceed payload or structural limitations. Additionally, bang-bang control uses large amounts of propellant. As structural flexibility increases, minimizing modal vibration induced by on-off thrusters becomes more difficult for the designer.

B. MOTIVATION FOR RESEARCH

1. Active versus Passive Control

Numerous approaches for dealing with flexibility have been researched, to varying degrees of success. Passive methods such as gain stabilization are used where attitude pointing requirements are less stringent than structural loads requirements. Active damping systems such as piezoelectric actuators show promise. However, complications with sensor/actuator collocation, real-time parameter estimation, and robustness persist.

2. Naval Postgraduate School Research Efforts

Using the Flexible Spacecraft Simulator (FSS) experimental facility, researchers at the Naval Postgraduate School Spacecraft Research and Design Center (SRDC) are studying vibration suppression during flexible spacecraft slewing maneuvers. One goal of this research is to realize a synergistic integration of maneuvering control actuators with active vibration control systems. To accomplish this goal, parallel efforts are addressing vibration suppression and vibration avoidance.

Active vibration controllers which work independent from slewing controllers focus on end-of-maneuver performance. SRDC researchers have completed several experiments in this area. Previous research includes implementations of Linear Quadratic Gaussian, Positive Position Feedback (PPF), and velocity feedback controllers with piezoceramic actuators bonded to a flexible beam. In order to understand how various slewing and vibration suppression techniques interact, SRDC researchers are also investigating control-structure interactions. Research effort into this phase concentrates on the slewing actuators which are typically a major vibration source. By pursuing a two stage strategy, improved solutions to vibration control will likely employ the best characteristics of several techniques. Capitalizing on the strengths of different methods can greatly ease the demands on each sub-system without incurring excessive complexity. For example, an active piezoceramic vibration suppression system integrated with a shaped slewing command may virtually eliminate all modal vibrations.

3. Potential Impact of Input Shaping Techniques

Input shaping for systems with linear actuators has been successfully developed to act on modal vibrations as they occur. Recently, this method has been extended to systems with on-off actuators to some degree. However, existing approaches require complicated non-linear optimization and result in bang-bang control action. The principal drawbacks stem from an extensive set of optimization constraints which, in the presence of multiple, closely spaced modes, are extremely sensitive to initial state, number of modes, mode ratio, and move distance. Clearly, an approach which capitalizes on the strengths of command input shaping without carrying the drawbacks is preferable.

C. SCOPE OF THESIS

This thesis presents the first study of Pulse-Width, Pulse-Frequency (PWPF) modulated thruster control using the method of variable amplitude command input shaping. Prior to implementing the input shaping technique, the PWPF modulator is studied and design guidelines are listed for the first time. The efficacy of PWPF modulated thruster control with variable amplitude input shaping is demonstrated by computer simulations. This research provides the requisite analytical framework for future vibration control experiments involving the FSS.

The current FSS model is extended to include piezoceramic sensors and actuators mounted on the flexible appendage. After the equations of motion for the model are determined, an in-depth analysis of the PWPF modulator is conducted to determine the tunable range on modulator parameters. Recommend settings are included. Next, variable amplitude command input shapers are designed and integrated with the PWPF modulator. Investigations of single-mode performance are extended to multiple-mode cases and comparisons made between shaper type and targeted modes. Robustness analyses are then performed to ensure viability of the approach with uncertain plant conditions.

This thesis will show that the integration of a PWPF modulator with input shaping techniques provides a simple way of avoiding modal vibrations for flexible spacecraft with on-off actuators. Prior research which has shown the superiority of PWPF techniques over bang-bang control in terms of thruster cycle and propellant economy will be extended to include vibration-free maneuvers. Finally, this thesis is written as a reference for future research efforts into the area of modulated thrusters. Since much of the information on PWPF resides in corporate technical memoranda and not in the published literature, this thesis provides a convenient reference for follow-on researchers.

II. FLEXIBLE SPACECRAFT SIMULATOR (FSS) SYSTEM MODEL

A. EXPERIMENTAL FACILITY

The FSS experiment at the Naval Postgraduate School was designed to investigate the effects of various control schemes on the slewing vibration suppression performance of flexible spacecraft. The experimental facility, depicted in Figure 2.1, is a two-dimensional model of a typical spacecraft. The FSS consists of a 0.76-meter diameter rigid central body (hub) and a flexible, "L"-shaped beam fixed to the hub perimeter. The 2.22-cm thick central body is restricted from translational motion by means of an air-bearing support structure. In order to simulate a frictionless environment, both the central body and flexible beam are floated on air pads. The FSS is equipped with a momentum wheel and 0.35N cold gas thrusters for executing attitude control maneuvers. Complete descriptions of the experiment, including discussions on real-time control and data collection, can be found in Watkins (1991), Hailey (1992), and McClelland (1994).

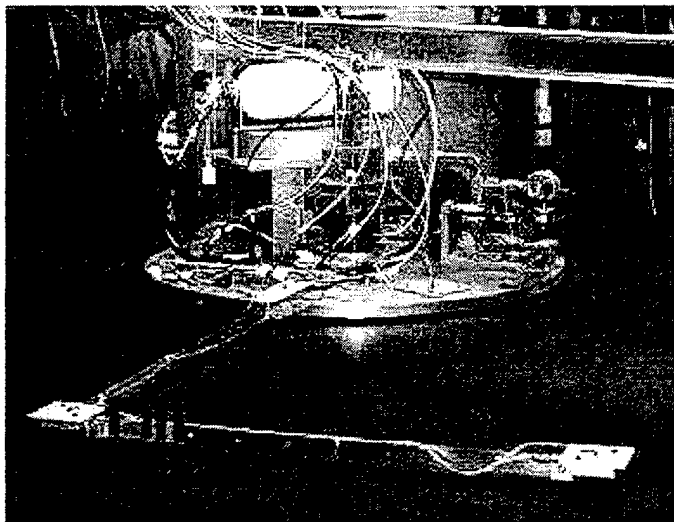


Figure 2.1
NPS FSS Experiment

B. FSS FLEXIBLE APPENDAGE

1. Description

The flexible appendage, termed the “arm”, is used for researching active vibration control schemes and is fully reconfigurable. Piezo-ceramic (PZT) sensor/actuator patches can be applied to the beam to control vibration and arm shape. Figure 2.2 shows a typical flexible appendage configuration with PZT patches at the beam elbow and base.

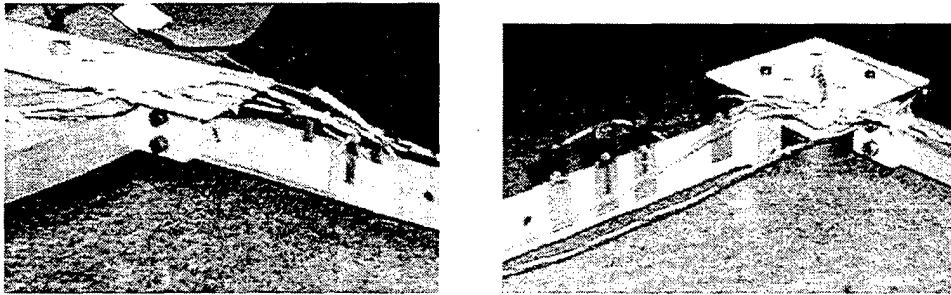


Figure 2.2

Base joint (left) and elbow joint (right) with piezoceramic patches and LED Targets

The beam used in this analysis consists of two 0.61-m long aluminum strips joined at ninety degrees by a bracket. PZT sensors and actuators are included at the base of the arm and elbow. Table 2.1 lists the properties of the beam.

Table 2.1
FSS Flexible Beam Properties

| Property (units) | Value |
|---------------------------------------|-----------------------|
| Beam thickness, <i>mm</i> | 1.58 |
| Beam height, <i>cm</i> | 2.54 |
| Beam density, <i>kg/m³</i> | 2.80×10^3 |
| Beam Modulus, <i>N/m²</i> | 7.20×10^{10} |

Mass intensifiers, known as “point masses”, are placed at various locations along the length of the arm in order to reduce the fundamental cantilever frequency to approximately 0.18 Hz. Using this setup, large structures with low fundamental frequencies can be modeled using a scaled experimental facility.

2. Finite Element Model

An eight element model of the flexible arm as configured for this investigation was constructed using MATLAB. All motion was considered to be in-plane bending based on a cantilevered mounting. Torsional effects were not included. Structural damping was assumed to be 0.4% ($\zeta = 0.004$) for all modes. In order to reduce the cantilever frequencies, point masses were added at each node as shown in Table 2.2.

Table 2.2
Finite Element Model Nodal Mass Distribution

| Node ¹ | Point Mass (kg) | Piezo Sensor/Actuator ² |
|-------------------|-----------------|------------------------------------|
| 1 | 0.455 | Element 1 |
| 2, 3 | 0.455 | none |
| 4 | 0.91 | none |
| 5 | 0.455 | Element 5 |
| 6, 7 | 0.455 | none |
| 8 | 0.91 | none |

Notes: (1) The base of the arm is defined as node zero.

(2) Elements span the listed node and the one prior.

PZT sensors and actuators are mounted at the elbow and base of the arm. Addition of the piezo patches adds beam stiffness and therefore increases the fundamental frequency. However, addition of the point masses overcomes this effect and significantly lowers the beam natural frequencies. Table 2.3 lists the piezoceramic properties used in developing the finite element model. The fundamental cantilever frequency is 1.15 rad/sec (0.18 Hz). A complete listing of the cantilever modes is included in the following section.

Table 2.3
Piezoceramic Sensor/Actuator Properties

| PROPERTY | SYMBOL | VALUE |
|---|--------------|------------------------|
| Piezo Lateral Strain Coefficient, m/V | d_{31} | 1.80×10^{-10} |
| Piezo Modulus, N/m^2 | E_p | 6.30×10^{10} |
| Piezo Permittivity, N/V^2 | ϵ_3 | 1.50×10^{-8} |
| Piezo sensor thickness, mm | t_{ps} | 0.25 |
| Piezo actuator thickness, mm | t_{pa} | 0.50 |
| Piezo density, kg/m^3 | ρ_p | 7.70×10^3 |

C. EQUATIONS OF MOTION

The linearized equations of motion for the FSS equipped with thrusters (T_i) and momentum wheel (T_w) have been developed in Watkins (1991) and Hailey (1992). However, addition of piezoceramics to the beam adds the piezo voltage as an additional state as well as another independent equation. The equations of motion for the flexible spacecraft without piezos are given by:

$$I_{zz}\ddot{\theta} + I_w\ddot{\theta}_w + \sum_{i=1}^n D_i\ddot{q}_i = T_C + T_D \quad (2.1)$$

$$I_w\ddot{\theta} + I_w\ddot{\theta}_w = T_w \quad (2.2)$$

$$\ddot{q}_i + D_i\ddot{\theta} + Z\dot{q}_i + \lambda q_i = 0 \quad (2.3)$$

where equation (2.1) represents the rigid-body motion and its coupling to the flexible beam and momentum wheel. Equation (2.2) represents the momentum wheel equation and shows the wheel coupling to the rigid-body. Equation (2.3) is the flexible beam equation showing coupling to the rigid-body mode. The variables and symbols used in these equations are defined as follows:

| | |
|------------|--|
| I_{zz} | Flexible spacecraft moment of inertia |
| I_w | Momentum wheel inertia |
| θ | Central body angular position |
| θ_w | Momentum wheel angular deviation |
| D_i | Rigid-elastic coupling vector, comprised of components D_i |
| T_C | Control torques, $T_C = T_w + T_t$ |
| T_D | Disturbance torques |
| T_w | Momentum wheel torque |
| q_i | Flexible arm node displacements |
| Z | Modal damping matrix, $2\zeta_i\omega_i$ (diagonal) |
| λ | Matrix of natural frequencies, ω_i^2 (diagonal) |

1. Rigid-Elastic Coupling Effect

Equations (2.1) - (2.3) were obtained by writing expressions for the kinetic and potential energy of the rigid body and flexible beam. There are three sources of kinetic energy due to rigid body slewing: 1) the rigid body rotation, 2) flexible beam cantilever motion, and 3) the combined translation and rotation ($\omega \times r$ terms) of the flexible beam. The third source of kinetic energy is the means by which the rigid body and flexible responses couple. The rigid-elastic coupling vector, D , is a measure of this interaction. D has dimension $n \times 1$ where n is the number of flexible modes. A detailed derivation of the rigid-elastic coupling term is presented in Agrawal (1996). The rigid-elastic coupling for each vibrational mode is given by

$$D_i = \int_F (x_F \phi_i^y - y_F \phi_i^x) dm \quad (2.4)$$

where x_F and y_F are the coordinates of each finite element node and the ϕ_i 's are x and y displacements, respectively, of each node from the unperturbed position. The integration over mass weights the nodes according to the amount of lumped mass at each location. Calculating the rigid-elastic coupling using data from a finite element modeling program is performed using:

$$D_i = \sum_{j=1}^N (x_F \phi_i^{yj} - y_F \phi_i^{xj}) m_j \quad (2.5)$$

where there are N finite element nodes. The lumped mass parameter, m_j , includes the mass from the beam element at each node as well as any point masses mounted at the node. The calculations are repeated for each of the n modes. An example calculation of the first mode rigid-elastic coupling term, D_1 , is provided in Appendix A.

2. Addition of Piezoceramics to the FSS Model

Addition of piezo-ceramic sensors and actuators modifies the original equations by adding stiffness to the beam and applying reaction torques to beam bending. The voltage measured by a piezoceramic sensor is proportional to the amount of tension or compression imposed upon the structure. Using the piezoceramic sensor voltage as a generalized coordinate and evaluating the system Lagrangian, the operating equation, or “sensor equation”, for piezo-ceramic actuators is (Meyer & Agrawal, 1996):

$$e = \frac{1}{\gamma} \mathbf{B}^T \mathbf{q} \quad (2.6)$$

The electro-mechanical coupling term, \mathbf{B}^T , represents the conversion of electrical voltage, e , to mechanical displacement, q at each node and is defined as

$$\mathbf{B}^T = [b_1 \quad b_2 \quad b_3 \quad b_4] \quad (2.7)$$

where γ , b_i , and e are constants determined by the material properties of the beam and the piezoceramic patches. In general, the electrical voltage may correspond to an actuator input or a sensor output. Actuator voltages are expressed as e_a and sensor voltages as e_s . The voltage is considered constant across the piezoceramic cross-section and is given by

$$e = t_p E$$

Expressions developed in Agrawal (1996) describing the piezoceramic contribution to the model are listed here for reference:

$$\begin{aligned}
\gamma &= \frac{w_p h}{t_p} \left(\epsilon_3^T - d_{31}^2 E_p \right) \\
b_1 &= 0 \\
b_2 &= -d_{31} E_p w_p \left(\varsigma + \frac{t_p}{2} \right) \\
b_3 &= 0 \\
b_4 &= d_{31} E_p w_p \left(\varsigma + \frac{t_p}{2} \right)
\end{aligned} \tag{2.8}$$

where the variables are described as follows:

| | |
|-------------|---|
| w_p | Width of the piezoceramic patch |
| h | Height of the piezoceramic patch |
| t_p | Thickness of the piezoceramic patch |
| ς | Distance of beam surface from coordinate axis |

The equation of motion for the piezoceramic response to an actuator input is simply

$$M_p \ddot{\mathbf{q}} + \mathbf{K}_p \mathbf{q} = -\mathbf{B} e_a \tag{2.9}$$

Eq. (2.9) is of the same form as the flexible beam equation so the piezoceramic elemental mass and stiffness matrices may be added directly to the beam elemental matrices prior to conversion to global coordinates. The resulting equations of motion for the FSS with thrusters, momentum wheel, and piezoceramic sensors/actuators are

$$\begin{aligned}
I_{zz} \ddot{\theta} + I_w \ddot{\theta}_w + \sum_{i=1}^n D_i \ddot{\mathbf{q}}_i &= T_C + T_D \\
I_w \ddot{\theta} + I_w \ddot{\theta}_w &= T_w
\end{aligned} \tag{2.10}$$

$$\ddot{\mathbf{q}}_i + \mathbf{D} \ddot{\theta} + \mathbf{Z} \dot{\mathbf{q}}_i + \lambda \mathbf{q}_i = -\mathbf{B} e_a$$

$$\gamma e = \mathbf{B}^T \mathbf{q}$$

Note that the piezo actuator voltage term, $\mathbf{B}e_a$, can describe a state-feedback controller or an externally applied voltage for shape or feedforward vibration control. For example, if a velocity feedback control law is implemented with piezoceramic actuators, e_a can be rewritten in the following form:

$$e_a = -k_d \dot{e}_s = \frac{-k_d}{\gamma} \mathbf{B}_s^T \dot{\mathbf{q}} \quad (2.11)$$

3. Model Configuration

Since this thesis focuses on thruster control, the FSS momentum wheel is not utilized. Therefore, θ_w and T_w are taken to be zero and the wheel moment of inertia is included in the total moment of inertia $I_{zzo} = I_{zz} + I_w = 10.49 \text{ kg-m}^2$. Disturbance torques are neglected in order to isolate the effects of the thruster on the flexible simulator. At user discretion, a velocity feedback controller is included to provide piezo actuator control torques. The finite element code generates system matrices for all configurations and queries the user for the set to be used in the analysis.

Up to sixteen flexible modes may be selected in the finite element model code. Based on the relative frequencies, a maximum of eight modes is used in this thesis without significant penalty in computation time or complexity. Using beam configuration listed above and the material properties listed in tables 2.1 and 2.3, the piezoelectric parameters are

$$\mathbf{B}_a^T = \begin{bmatrix} 0 & -3.708 \times 10^{-4} & 0 & 3.708 \times 10^{-4} \end{bmatrix}$$

$$\mathbf{B}_s^T = \begin{bmatrix} 0 & -2.628 \times 10^{-4} & 0 & 2.628 \times 10^{-4} \end{bmatrix}$$

$$\gamma_a = 1.0033 \times 10^{-7}$$

$$\gamma_s = 2.0065 \times 10^{-7}$$

and the rigid-elastic coupling vector, \mathbf{D} , is

$$\mathbf{D} = \begin{bmatrix} -1.687 & -1.185 & 0.174 & 0.224 & 0.066 & 0.144 & 0.037 & 0.088 \end{bmatrix}^T$$

The stiffness and damping matrices λ and Z of equation (2.10) are block diagonal:

$$\lambda = \begin{bmatrix} \omega_1^2 & & 0 \\ & \omega_1^2 & \\ 0 & & \ddots \\ & 0 & & \omega_n^2 \end{bmatrix} \quad \text{and} \quad Z = \begin{bmatrix} -2\zeta_1\omega_1 & & 0 \\ & -2\zeta_2\omega_2 & \\ 0 & & \ddots \\ & 0 & & -2\zeta_n\omega_n \end{bmatrix} \quad (2.12)$$

where ω_n 's are the cantilever frequencies of the flexible arm. Table 2.4 lists the cantilever and system natural frequencies.

Table 2.4
Flexible Spacecraft Simulator Modal Frequencies

| Mode | Cantilever Frequency | | System Frequency | |
|------|----------------------|-------|------------------|-------|
| | (rad/sec) | (Hz) | (rad/sec) | (Hz) |
| 1 | 1.150 | 0.183 | 1.34 | 0.213 |
| 2 | 2.840 | 0.452 | 3.16 | 0.504 |
| 3 | 15.20 | 2.41 | 15.23 | 2.42 |
| 4 | 26.61 | 4.23 | 26.72 | 4.25 |
| 5 | 52.92 | 8.42 | 52.94 | 8.42 |
| 6 | 77.18 | 12.3 | 77.31 | 12.3 |
| 7 | 104.2 | 16.6 | 104.2 | 16.6 |
| 8 | 132.0 | 21.0 | 132.1 | 21.0 |

D. STATE SPACE FORMULATION

Equations (2.10) can be written directly as a partitioned second order system given by

$$\begin{bmatrix} I_{zz} & D^T \\ D & \Xi \end{bmatrix} \begin{Bmatrix} \ddot{\theta} \\ \ddot{q} \end{Bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & Z \end{bmatrix} \begin{Bmatrix} \dot{\theta} \\ \dot{q} \end{Bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & \lambda \end{bmatrix} \begin{Bmatrix} \theta \\ q \end{Bmatrix} = \begin{bmatrix} T_T \\ -2Be_a \end{bmatrix} \quad (2.13)$$

where the total rigid body torque $T_T = T_i + T_w + T_D$ is the sum of all torques acting on the body and the piezoelectric torque is due to a state feedback controller or an externally

applied voltage. A factor of two is applied to the piezoelectric torque to account for actuators placed on each side of the beam. The state space model of the form

$$\begin{aligned}\dot{x} &= Ax + Bu \\ y &= Cx + Du\end{aligned}$$

is obtained through a series of state transformations. Setting D (the direct transmission matrix) to zero, the resulting modal system is

$$\begin{Bmatrix} \dot{\eta} \\ \ddot{\eta} \end{Bmatrix} = \begin{bmatrix} 0 & \Xi \\ \Lambda & Z + Z_p \end{bmatrix} \begin{Bmatrix} \eta \\ \dot{\eta} \end{Bmatrix} + \begin{bmatrix} 0 \\ \Phi^T \end{bmatrix} \mu \quad \text{where } \mu = \begin{bmatrix} T_t \\ T_p \end{bmatrix} \quad (2.14)$$

where Z is defined in Eq. (2.12) above and Λ , similar in form to λ in Eq. (2.12), is the block diagonal partition of squared *system* natural frequencies. Note that the matrix Z can be comprised of unaugmented beam modal damping as well as active damping from the piezoelectric state feedback controller.

III. PULSE WIDTH, PULSE FREQUENCY MODULATORS

A. BACKGROUND

While PWPF modulators have been in use since the 1960's, little has been published concerning their performance or optimization. Considerable effort has been put into determining the stability margins of the modulator. However, analyses of varying modulator parameters for a given plant and mission are rare.

B. DESCRIPTION OF PWPF MODULATOR

The PWPF modulator is designed to provide a thruster output proportional to command input. The modulator improves pointing accuracy for all-thruster control and results in more efficient propellant use. The modulator, as illustrated in Figure 3.1, incorporates a first order lag filter and Schmidt trigger with a feedback loop. The lag filter integrates the error between command and actuator state and is tunable to achieve the desired system response. Assuming a zero initial condition, the thruster fires once the trigger threshold, d , is exceeded. The thruster will continue firing until the Schmidt trigger input falls below the off-threshold, h . Addition of a pre-filter input gain, K_p , upstream of the summation allows the designer to ensure command inputs will remain in the pseudo-linear range as well as to tailor the response as desired.

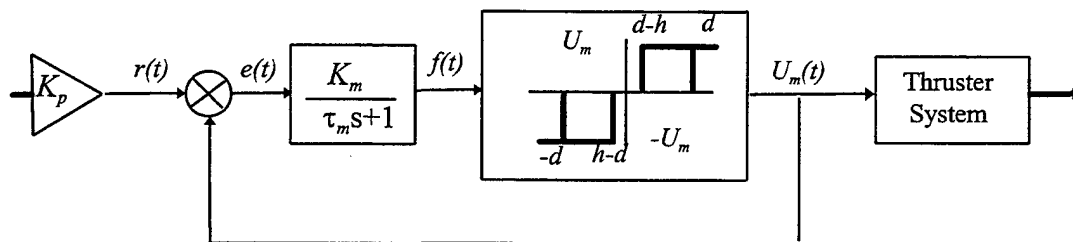


Figure 3.1
PWPF Modulator

Nomenclature for the PWPF modulator parameters shown in Figure 3.1 are listed below:

| | |
|----------|---|
| $r(t)$ | input command |
| $U_m(t)$ | modulator output, magnitude of U_m |
| $e(t)$ | command error signal, $r(t) - U_m(t)$ |
| $f(t)$ | filter output signal to thruster |
| d | thruster on-threshold, also termed " E_{on} " |
| h | thruster off-threshold, also termed " E_{off} " |
| $d-h$ | hysteresis parameter |
| K_m | modulator gain |
| K_p | pre-filter gain |
| τ_m | pre-filter time constant |

1. Motivation for PWPF Modulated Slewing

PWPF modulated thrusters allow acceptable slew performance with less vibration than a simple bang-bang control (McLelland, 1994). The superior performance of the PWPF modulated control to the other systems is a combination of three factors.

First, the PWPF modulator approximates a linear actuator by varying both pulse width and frequency with time. Operation in this linear range, known as "pseudo-linear" operation, gives the designer a response which is analogous to that of a linear actuator such as a momentum wheel.

Second, because the controller has the option to vary the pulse size and timing, the PWPF modulator is more fuel efficient than the bang-bang controller, especially in the presence of flexibility. Because the bang-bang controller is unable to produce a linear output, thruster firings often excite flexible modes which then couple back in to the rigid body motion. The result is a limit cycle with continuous thruster switching and propellant waste. The PWPF controller, besides causing fewer vibrations, can tailor its response to reduce the number of thruster firings.

Finally, the PWPF controller offers flexibility in design and operation. PWPF modulator components can be tuned to optimize performance for any given plant configuration, for example, when spacecraft moments of inertia change over the course of a mission. Because the modulator parameters are independent of the spacecraft parameters, no *a priori* knowledge of the spacecraft dynamics is required for system

analysis and performance determination. This characteristic provides the scope for implementing adaptive- or fuzzy control schemes.

2. Previous Research into PWPF Modulation

For all of its advantages, few applications of PWPF in conjunction with other vibration control schemes have been reported in published literature. Several complications caused by the modulator may be the primary reason for its limited exposure. The primary disadvantage of the PWPF modulator is its tendency to contribute to phase lag at higher frequencies (Wie and Plescia, 1984; McClelland, 1994). Additionally, its inherent nonlinear characteristics make precise stability margin determinations difficult. Anthony and Wie (1990) attempted to quantify the modulator stability margins using various describing functions with limited success. One of the best contributions from this research is the PWPF modulator module to NASA's Interactive Controls Applications (INCA) software. In general, there is very little that can be quantitatively applied across the board to all modulator configurations. Rather, the modulator must be tuned for desired response and then checked for stable operation at that operating point. In the event a particular set of modulator parameters is unsuitable, even a small change in a parameter can dramatically change the stability margins (Anthony and Wie, 1990). Advances in computing power and the development of improved describing function methods simplify these analyses.

Complicated nonlinear stability analyses should not deter use of the PWPF modulator, however. Complete knowledge of the stability margins for all modulator settings is unnecessary. Rather, the designer must determine the specific parameter settings needed to attain desired system performance and check the stability of that unique case.

C. PWPF TIME DOMAIN EQUATIONS

The equations governing the PWPF modulator provide insight into the effects of parameter choice on system performance and limits. All derivations were conducted as part of the NPS course AA4900 "Thruster Control of Flexible Spacecraft" (Meyer, 1995).

1. Modulator Time Response

Referring to Figure 3.1, the error signal is

$$e(t) = r(t) - U_m(t) \quad (3.1a)$$

Or in Laplace form

$$e(s) = r(s) - U_m(s) \quad (3.1b)$$

The filter output is given by

$$f(s) = \frac{K_m}{\tau_m s + 1} [r(s) - U_m(s)] + \frac{\tau_m f(0)}{\tau_m s + 1} \quad (3.2)$$

let $r(t)$ be a commanded step input of arbitrary magnitude A , such that

$$r(s) = \frac{A}{s} \quad (3.3a)$$

The controller output $U_m(t)$ is a step function as well:

$$U_m(s) = \frac{U_m}{s} \quad (3.3b)$$

so that the filter output becomes

$$\begin{aligned} f(s) &= \frac{K_m}{\tau_m s + 1} \left[\frac{A}{s} - \frac{U_m}{s} \right] + \frac{\tau_m f(0)}{\tau_m s + 1} \\ &= \frac{K_m A}{s(\tau_m s + 1)} - \frac{K_m U_m}{s(\tau_m s + 1)} + \frac{\tau_m f(0)}{\tau_m s + 1} \end{aligned} \quad (3.4)$$

Partial fraction expansion of the first two terms yields

$$\begin{aligned} f(s) &= \frac{K_m A}{s} - \frac{K_m A \tau_m}{\tau_m s + 1} - \left[\frac{K_m U_m}{s} - \frac{K_m U_m \tau_m}{\tau_m s + 1} \right] \\ \text{or} \quad f(s) &= K_m A \left[\frac{1}{s} - \frac{1}{s + 1/\tau_m} \right] - K_m U_m \left[\frac{1}{s} - \frac{1}{s + 1/\tau_m} \right] \end{aligned} \quad (3.5)$$

such that the system time response is given by

$$f(t) = K_m A (1 - e^{-t/\tau_m}) - K_m U_m (1 - e^{-t/\tau_m})$$

or

$$f(t) = K_m (A - U_m) [1 - e^{-t/\tau_m}] \quad (3.6)$$

The initial condition response is obtained in a similar fashion:

$$f_i(s) = \frac{\tau_m f(0)}{\tau_m s + 1} \Rightarrow f_i(t) = f(0) e^{-t/\tau_m} \quad (3.7)$$

The total system response is given by:

$$f(t) = K_m (A - U_m) [1 - e^{-t/\tau_m}] + f(0) e^{-t/\tau_m} \quad (3.8)$$

2. System Operating Equations

The equations which express the behavior of the controller can be derived by analyzing the time domain equation between the values for the E_{on} , defined as “ d ”, and E_{off} , defined as “ $d-h$ ”, thresholds. When the Schmidt trigger is off, $U_m(t)=0$. With a zero initial output state, a reference step causes the filter output $f(t)$ to increase as a first order exponential. Once $f(t)$ reaches a value of d , the thruster fires with a magnitude $U_m(t)$ and reduces $f(t)$ as a first order exponential. The thruster is secured when $f(t)$ reaches $d-h$. The on-time, T_c , is the interval during which the thruster is activated (U_m non-zero). T_D , the off-time, is the interval between firings. As shown in Figure 3.2, the initial on-time and cycling on-time may differ, depending on the selection of Schmidt trigger parameters.

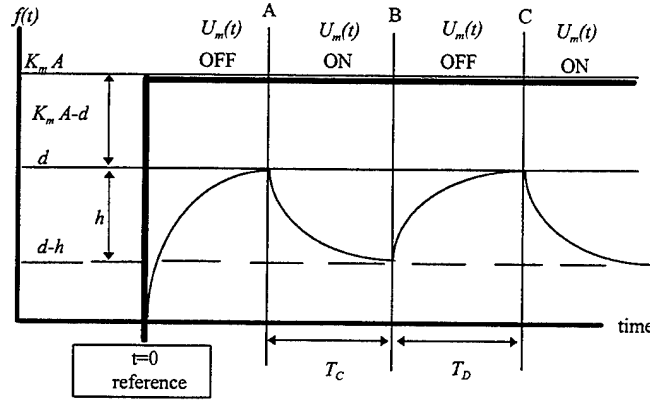


Figure 3.2
PWPF Lag Filter Time Response

a. On-time, T_c

The system response, Eq. (3.8), can be rewritten as:

$$f(t) = f(0) + (K_m A - f(0)) \left[1 - e^{-t/\tau_m} \right] \quad (3.9)$$

From point A to point B, $f(0)=d$ and $U_m(t)=U_m$. Referring to Figure 3.2, the difference in the filter output between off-on and on-off = $d-h$.

$$\Delta f_{on}(t) = d + \left[K_m (A - U_m) - d \right] \left[1 - e^{-T_c/\tau_m} \right] = d - h \quad (3.10)$$

Solving for T_c ,

$$e^{-T_c/\tau_m} = \frac{\left[K_m (A - U_m) - d \right] - h}{\left[K_m (A - U_m) - d \right]}$$

or

$$T_c = -\tau_m \ln \left[1 + \frac{h}{\left[K_m (A - U_m) - d \right]} \right] \quad (3.11)$$

b. Off-time, T_D

From point B to point C, $f(0)=d-h$ and $U_m(t)=0$. Referring once again to Figure 3.2,

$$\Delta f(t) = \left[(d-h) + (K_m(A-U_m) - (d-h)) \right] \left[1 - e^{-T_D/\tau_m} \right] - (d-h) = h \quad (3.12)$$

Solving for T_D ,

$$e^{-(T_D/\tau_m)} = \frac{h - [K_m(A-U_m) - (d-h)]}{-[K_m(A-U_m) - (d-h)]} \quad (3.13)$$

or

$$T_D = -\tau_m \ln \left[1 - \frac{h}{K_m(A-U_m) - (d-h)} \right] \quad (3.14)$$

c. Output frequency

The filter output frequency is defined as the inverse of the total period

$$f = \frac{1}{T_{total}} = \frac{1}{T_C + T_D} \quad (3.15)$$

Substituting for T_C and T_D ,

$$f = \frac{1}{-\tau_m \left[\ln \left(1 + \frac{h}{K_m(A-U_m) - d} \right) + \ln \left(1 - \frac{h}{K_m(A-U_m) - (d-h)} \right) \right]} \quad (3.16)$$

d. Modulation Factor

The modulation factor, also termed the duty cycle, is defined by the ratio of on-time to total period.

$$MF = \frac{T_C}{T_C + T_D} = \frac{1}{1 + T_D/T_C} \quad (3.17)$$

or in terms of the design parameters,

$$MF = \frac{1}{1 + \frac{-\tau_c \ln \left[1 - \frac{h}{K_m(A - U_m) - (d - h)} \right]}{-\tau_D \ln \left[1 + \frac{h}{K_m(A - U_m) - d} \right]}} \quad (3.18)$$

e. Minimum and Maximum Input Levels

The minimum input value, A_{min} , for which $T_c > 0$ is defined as the effective deadband of the modulator. Inputs smaller than the effective deadband will not trigger the relay, resulting in zero modulation factor. The minimum input for the relay to be on is

$$f_{min} = d - h = K_m(A - U_m) \quad (3.19)$$

which can be arranged as the fraction of modulation a given input represents as

$$\frac{K_m A - d}{K_m U_m - h} = \%MF \quad (3.20)$$

For an input A_{min} , equation (3.20) is identically zero, and the value of A_{min} is given by

$$A_{min} = \frac{d}{K_m} = \text{the effective deadband} \quad (3.21)$$

In the event a pre-filter gain, K_p , is utilized, the gain term in the denominator of equation (3.21) must be expressed as the product of the two gains, $K_p K_m$. Similarly, the maximum value of input A for which the modulator remains in pseudo-linear operation is found by setting equation (3.20) = 1 and solving for A_{max} .

$$\begin{aligned} \frac{K_m A_{max} - d}{K_m U_m - h} &= 1 \\ \Rightarrow A_{max} &= U_m + \frac{d - h}{K_m} \end{aligned} \quad (3.22)$$

f. Minimum Pulse Width

By definition, the on-time is the pulse width. Thus the minimum pulse width is the minimum on-time and is a function of K_m , h , and τ_m . Recall the on-time is

$$T_c = -\tau_m \ln \left[1 + \frac{h}{[K_m(A - U_m) - d]} \right] \quad (3.23)$$

Referenced to the turn-on time when $f(t) \geq f_{min}$ (d/K_m), the minimum pulse width becomes

$$\Delta = -\tau_m \ln \left[1 - \frac{h}{[K_m U_m]} \right] \quad (3.24)$$

Note that the minimum pulse width is a function of the hysteresis, the time constant, and filter gain (for a given set of command and thruster amplitudes). The time constant has a direct impact on phase lag of the system and as such should be as small as practically possible. However, as the time constant is reduced, the internal deadband is increased, resulting in declining performance. Hardware specifications will determine the minimum time constant as a function of the sampling period. If the equipment limitations preclude small enough hysteresis and/or τ_m , excessively large gains may be required in order to achieve the desired pulse width. If pulse width is excessively short, the controller will be driven beyond the linear range as the modulation factor approaches unity.

D. PWPF MODULATOR DESIGN ANALYSIS

Static and dynamic analyses of the PWPF modulator were conducted to define a set of design parameters which would provide the best all-around performance for use in slewing maneuvers of the flexible spacecraft. The principal objectives of the simulations were to verify the pseudo-linear operation, identify the tunable range of modulator parameters, and to justify parameter selections for the model. Prior to conducting the simulations, the PWPF time domain system equations were studied to determine the qualitative impact of varying design parameters. These observations were then quantitatively validated by simulating the response of the modulator to static inputs as

well as rigid body slewing commands. Simulations were run in batches, with sequentially varying PWPF parameters including: time constant, modulator gain, on-threshold, off-threshold, and pre-filter gain. The s-function "Schmidt.m" used to implement the Schmidt trigger in the SIMULINK system models is included in appendix B. Table 3.1 shows the range of parameters tested.

Table 3.1
PWPF Simulation Parameters

| Parameter | Range |
|-------------------------|---------------|
| Modulator Gain, K_m | 1.5 - 20 |
| Pre-filter Gain, K_p | 1.0 - 30 |
| Time Constant, τ_m | 0.01 - 1.0 |
| on-threshold, d | 0.0 - 1.0 |
| off-threshold, h | $0.0d - 1.0d$ |

1. Static Analysis

a. *Schmidt trigger parameters*

The Schmidt trigger is designed to reduce the number of thruster firings and total on-time through use of hysteresis and a deadband. Variation of the Schmidt trigger configuration can dramatically alter the system output. Prior to analyzing the operation of the modulator, a brief description of the Schmidt trigger parameters is in order.

(1) Off-threshold, h . This parameter sets the hysteresis value for a given deadband by defining the decreasing error signal amplitude for thruster cutoff. For a given on-threshold, d , increasing h decreases the hysteresis value while increasing the deadband. The impact is a decrease in frequency and an increase in minimum pulse width, resulting in a reduction in modulation factor.

(2) On-threshold, d . The trigger threshold sets the modulator deadband by defining the minimum error signal (fixed gain) which will result in thruster actuation. The modulator deadband is directly proportional to the value of d . Increasing

the on-threshold value increases the deadband and minimum pulse width, but does not impact the modulation factor or frequency.

(3) Controller output magnitude, U_m . Feedback of the modulator output reduces the error signal in order to control on and off operation of the thrusters. Increasing the modulator output level has an effect similar to reducing the modulator gain. As U_m increases, more time is required to reach both the on and off thresholds. The result is a decrease in frequency, an increase in minimum pulse width and a reduction in modulation factor. The internal deadband is unaffected by the modulator output level.

b. Pseudo-linear operation

One of the major advantages of the PWPF modulator is its characteristic pseudo-linear operation. Inspection of equation (3.18) reveals that for certain hysteresis values, modulator linearity is degraded for modulation factors close to unity and zero. The maximum acceptable input level for a given modulator configuration is given in equation (3.22) and repeated here for convenience.

$$\frac{K_m A_{\max} - d}{K_m U_m - h} = 1 \quad \Rightarrow \quad A_{\max} = U_m + \frac{d - h}{K_m}$$

To simplify the analysis, two design parameters can be defined. Let

$$B = \frac{K_m A_{\max} - d}{K_m U_m - h} \quad \text{and} \quad b = \frac{h}{K_m U_m - h} \quad (3.25a,b)$$

B is the fraction of the range of modulation and b is the fraction of hysteresis to the modulation range. Recasting the time domain equations in terms of B and b , the on- and off-time expressions are rewritten as

$$\begin{aligned} T_c &= \tau_m \ln \left(1 + \frac{b}{1 - B} \right) \\ T_D &= \tau_m \ln \left(1 + \frac{b}{B} \right) \end{aligned} \quad (3.26a,b)$$

Similarly, the frequency and modulation factor can be rewritten, yielding

$$f = \frac{1}{\tau_m \ln \left\{ \left(1 + \frac{b}{1-B} \right) \left(1 + \frac{b}{B} \right) \right\}}$$

$$\text{MF} = \frac{1}{\ln \left(1 + \frac{b}{B} \right) + \frac{1}{1 + \frac{b}{1-B} \ln \left(1 + \frac{b}{1-B} \right)}} \quad (3.27a,b)$$

Equation 3.27b is plotted as a function of b and B in Figure 3.3. Note the variations in linearity of modulation factor for a given fraction B as b varies from zero. In the limit as b tends to 0, the modulation factor is perfectly linear. In practice this cannot be realized without eliminating the hysteresis altogether. As b increases, the linearity breaks down in the regions of unity and zero MF. Note also that the off-threshold must always be smaller than the on-threshold. The design goal, therefore, is to select the largest non-zero off-threshold value, h , which allows linear operation while meeting hardware and performance limitations. The modulator time constant, τ_m , does not significantly impact the modulation factor or the pseudo-linear characteristics.

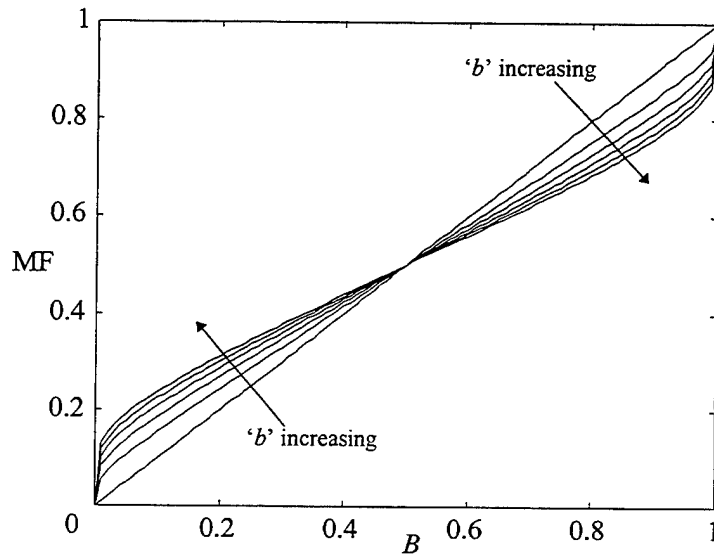


Figure 3.3
Impact of Modulator Parameters on Modulation Factor

c. Impact of design parameters on pseudo-linear operation.

Keeping this design guidance in mind, simulations were run to determine the modulation factor, thruster firing frequency, thruster cycles, and total thruster on-time for varying magnitude of the off-threshold as a function of on-threshold. Figures 3.4-3.7 show three trends.

(1) Zero hysteresis. As the off-threshold approaches the on-threshold value (hysteresis value $\rightarrow 0$), the number of thruster firings increases dramatically. At low modulator gains (< 6.0), the on-time is reduced despite the increase in firings, suggesting that each thruster cycle is extremely short in duration. In addition to non-linear operation, this situation will very likely violate hardware limitations and is undesirable due to the excessive cycles imposed over the life of the system. This condition is avoided by ensuring that the off-threshold is no more than approximately 80% of the on-threshold ($h \leq 0.8d$).

(2) Zero deadband. At the other extreme, Figure 3.6 shows that for modulator gain values above approximately $K_m = 4$, a very low on-threshold (< 0.1) with zero off-threshold results in rapidly increasing output frequency. For any combination of d and h , excessive gain exacerbates this effect and corresponds to eliminating the effective deadband from the system. Since this is counter to the design philosophy embodied in the Schmidt trigger, the effect is to put an upper bound on modulator gain and a lower bound on the on-threshold. There is very little change in the output frequency with varying on- and off-thresholds. In order to ensure there is sufficient deadband in the system, the lowest practical modulator gain (K_m) should be selected and the on-threshold should ensure pseudo-linear operation ($d \geq 0.3$ and $K_m \leq 6$). This combination of gain and on-threshold is a compromise between minimum gain requirements and modulator linearity. It results in an effective deadband of 0.05, which is well within the typical attitude control requirements of 0.1 degrees.

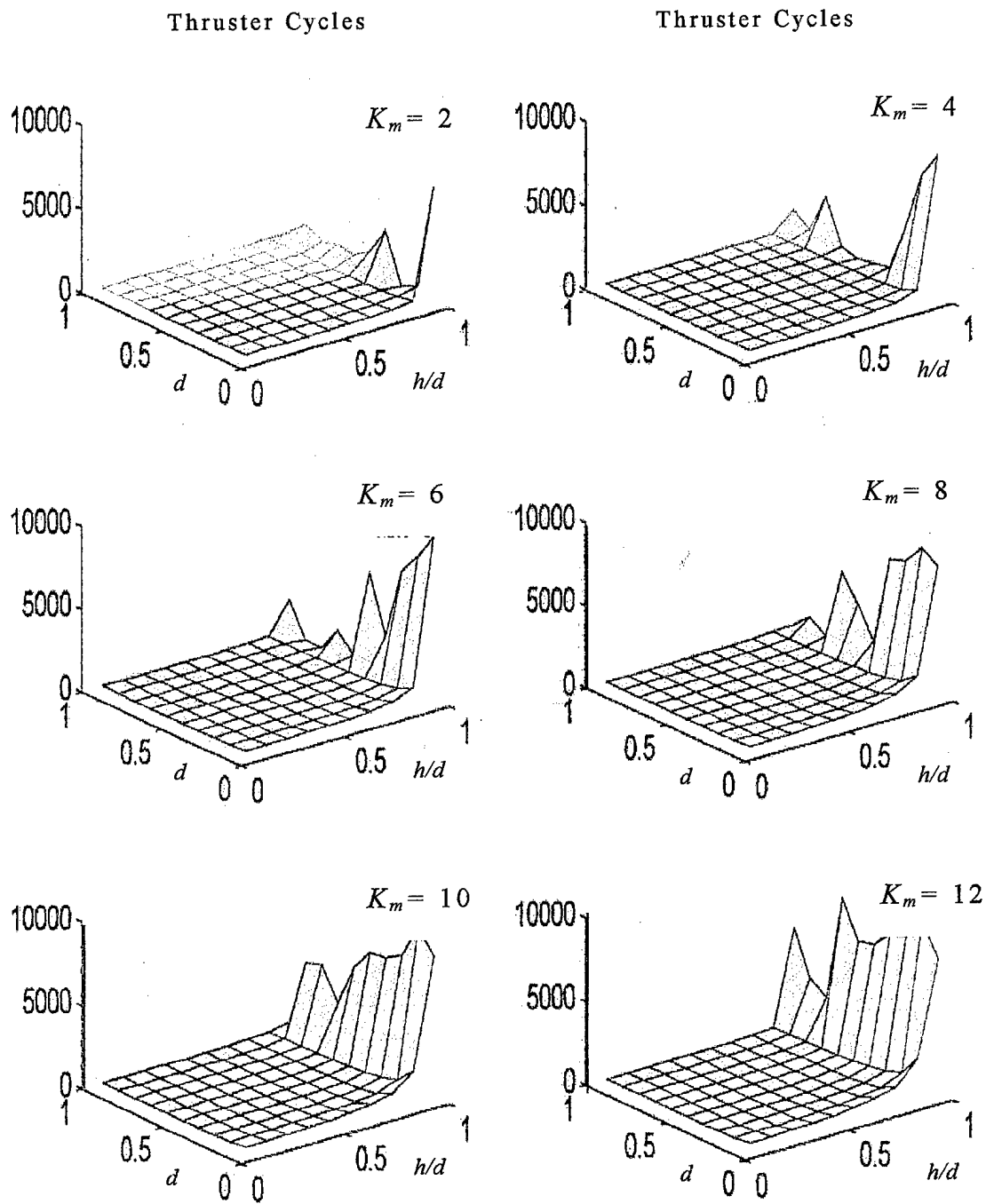


Figure 3.4
Variation of Thruster Cycles with Trigger Threshold

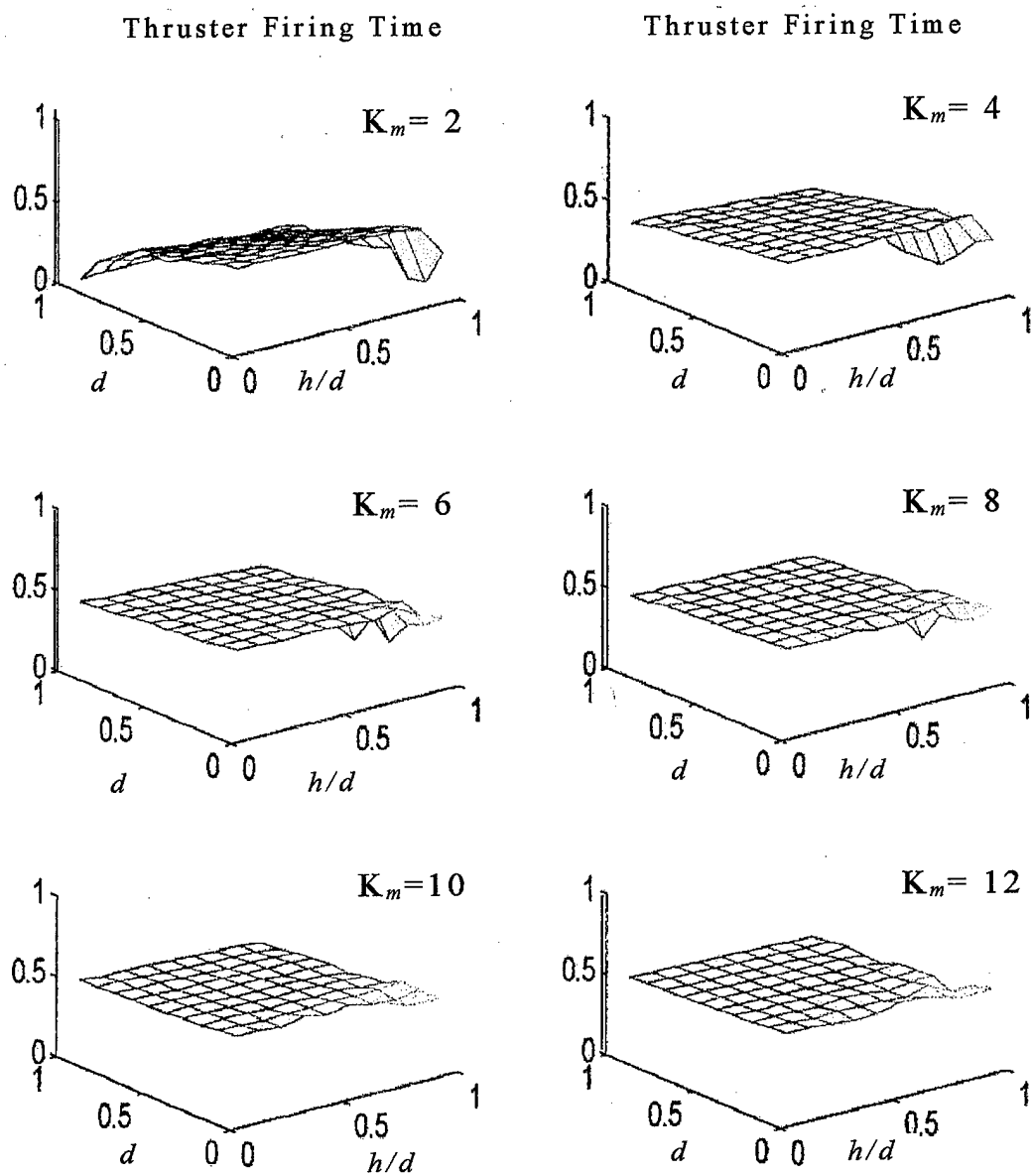


Figure 3.5
Variation of Thruster On-time with Trigger Threshold

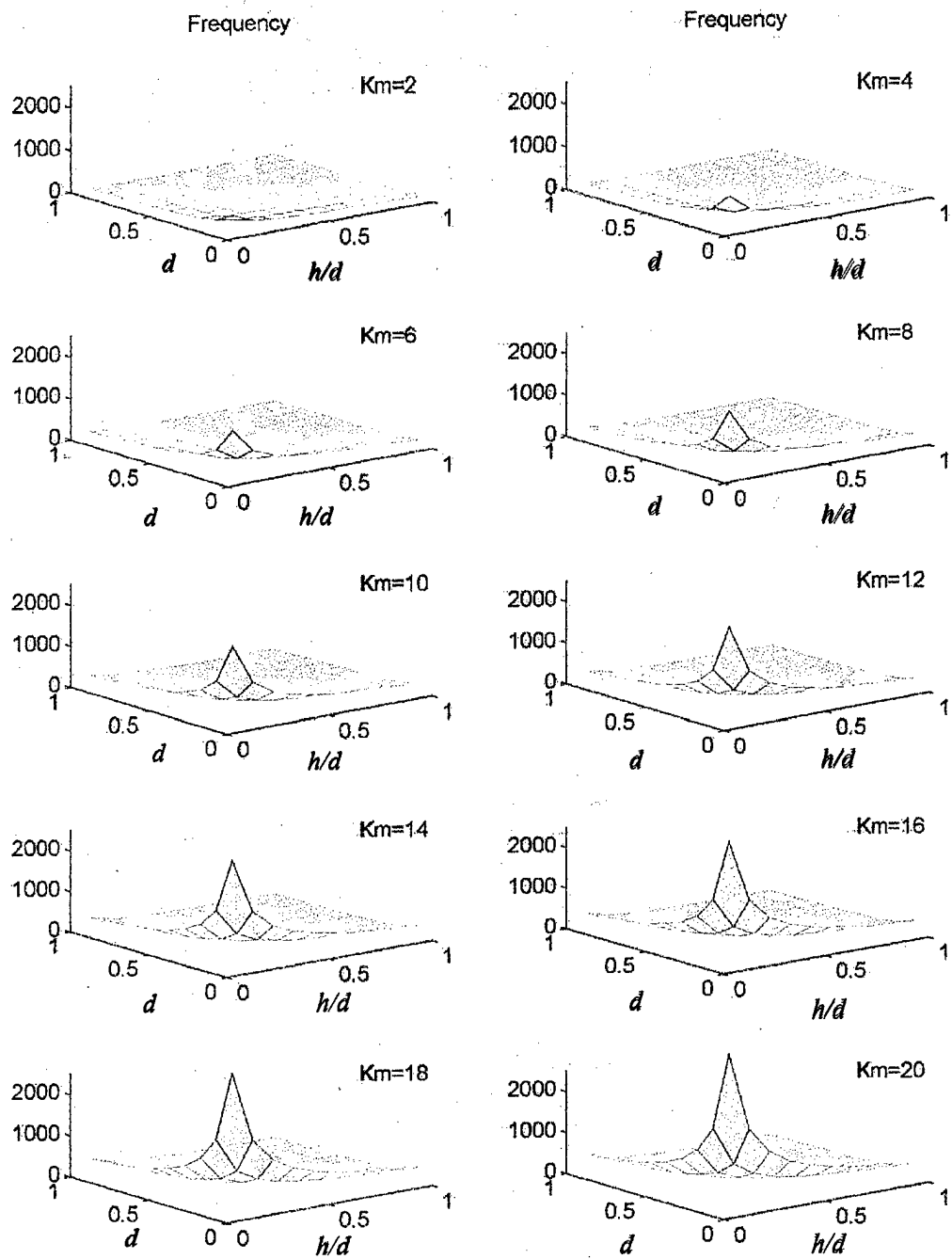


Figure 3.6
Variation of Thrusting Frequency with Trigger Threshold

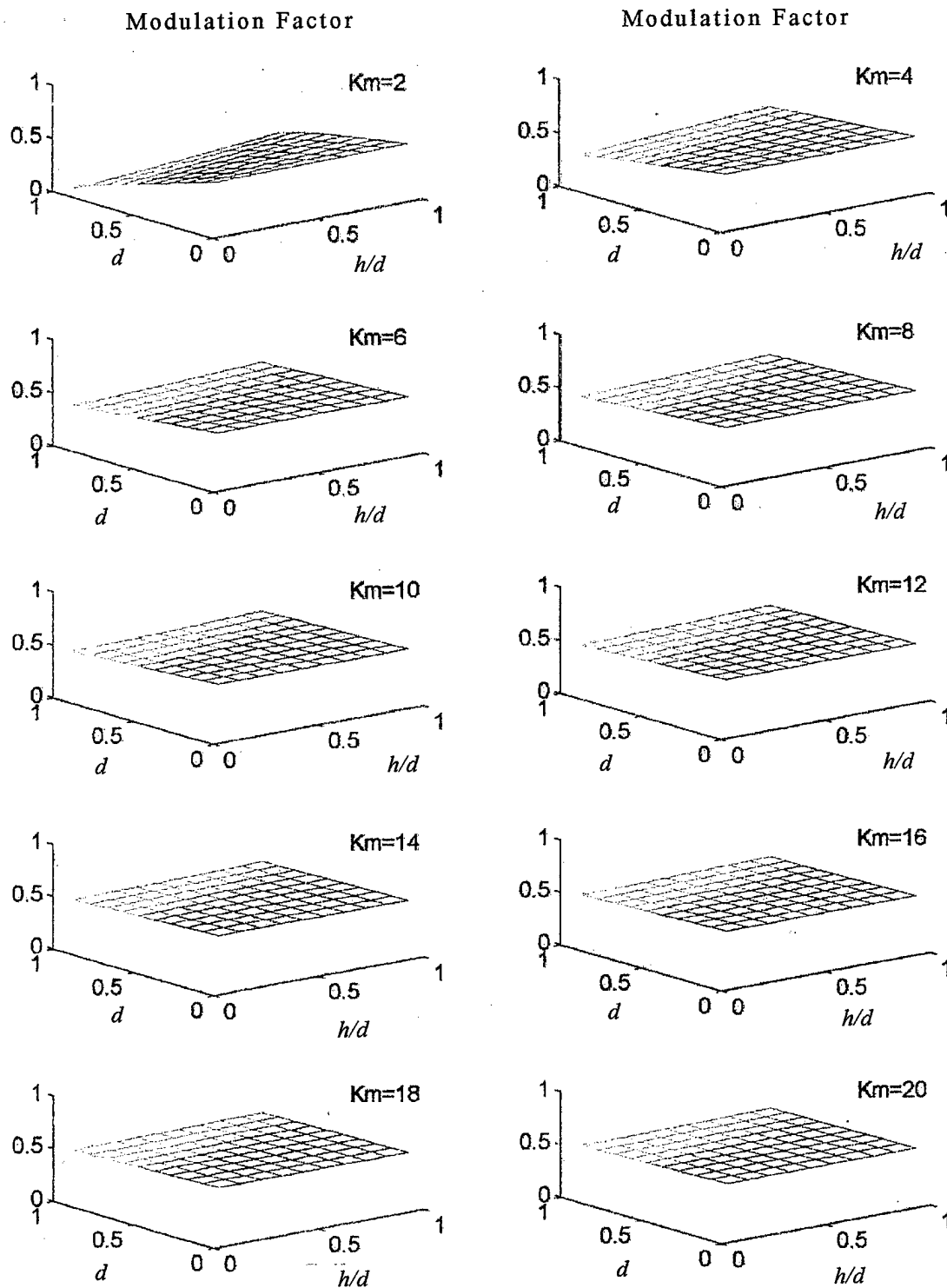


Figure 3.7
Variation of Modulation Factor with Trigger Threshold

(3) Limiting case for tailoring d and h . Figures 3.8 and 3.9 give further perspective on the relationship between the modulator gain and varying on- and off-thresholds.

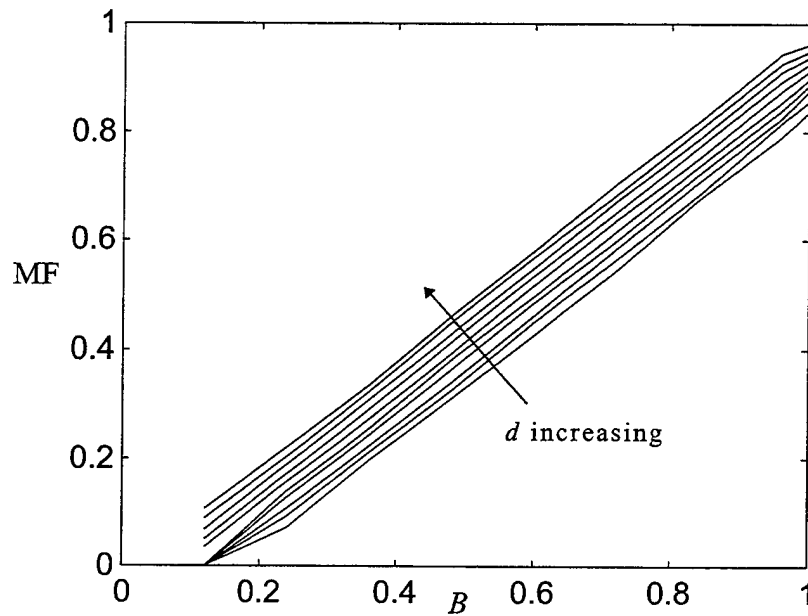


Figure 3.8
Impact of Trigger Thresholds on Modulation Factor

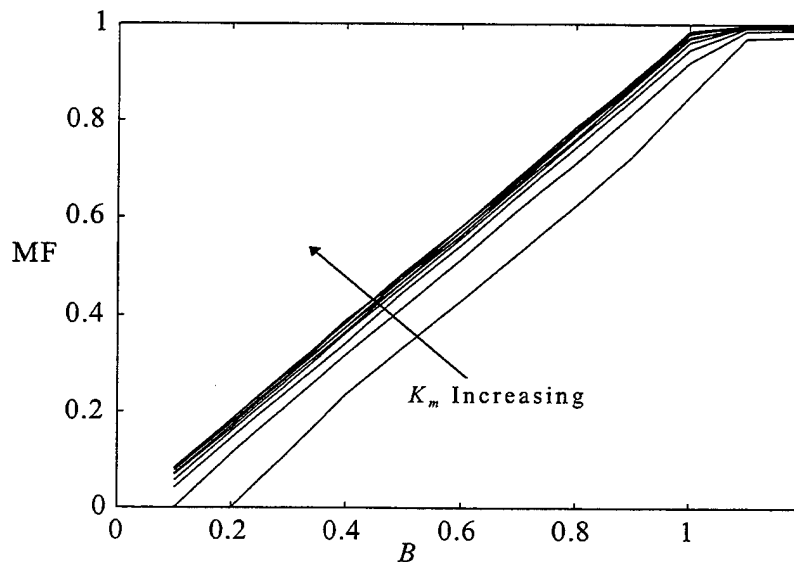


Figure 3.9
Variation of Modulation Factor with Modulator Gain, K_m

At low gain values, the modulation factor varies slightly with varying d and h . Operation in this area is still pseudo-linear. As K_m increases to approximately 6.0, choices of d and h have much less impact on the modulation factor. This limiting value is easily verified using equation 3.27b. The trend is echoed in Figure 3.10. Thus, at low modulator gain values, the hysteresis value can be tailored to effect changes in modulation factor. At K_m above approximately 4.0, hysteresis value has little impact on the modulation factor compared to K_m and modulator output level, U_m . The conclusion is that operation at low gain values is not as linear as at higher values, but that prudent selection of d and h allows pseudo-linear operation throughout the range of gains.

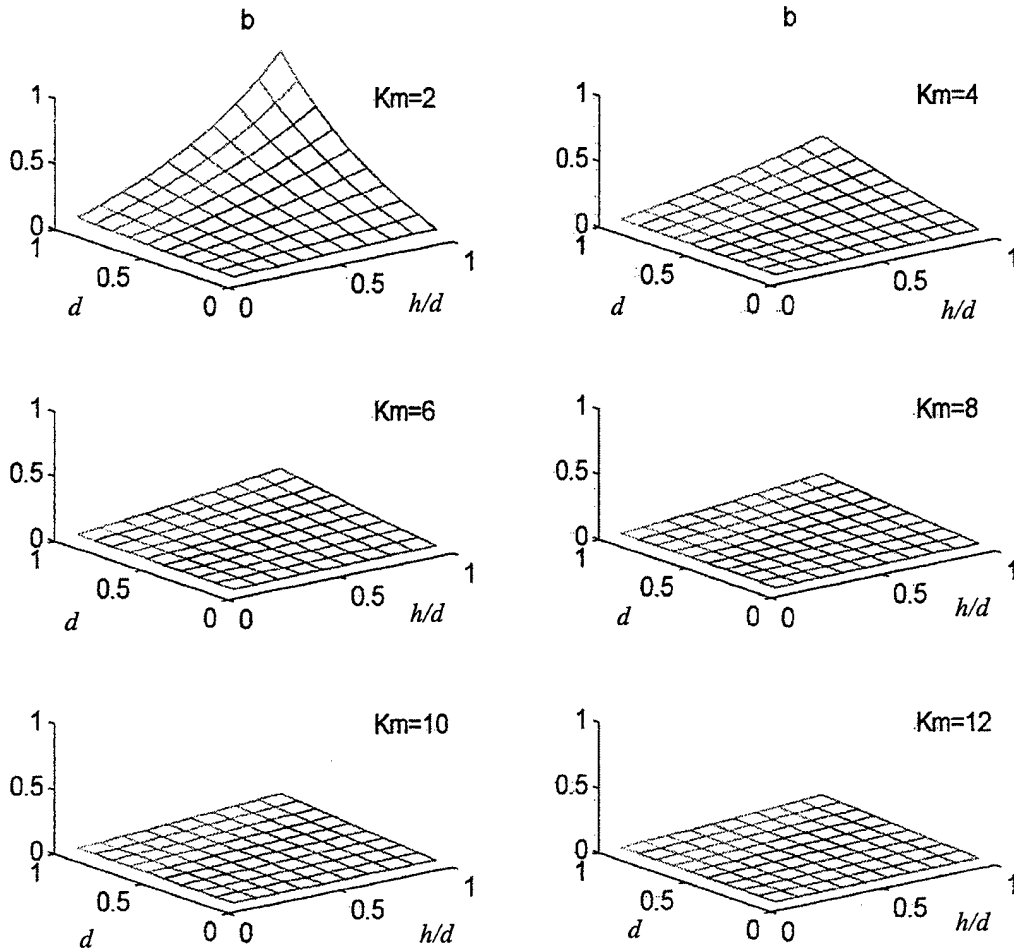


Figure 3.10
Variation of Modulation Fraction, b , with Trigger Thresholds

d. Effect of modulator gain, K_m

A large gain value may be desirable for noise and disturbance rejection, but must be tempered based on system response. As shown in Figure 3.9 (effect of K_m), at very low gain, the dead band decreases from 0.2 to 0.1 very quickly. Gain values greater than approximately 6.0 result in only small reductions in the deadband, but reduce the minimum pulse width and dramatically increase the output frequency. The number of thruster cycles becomes excessive. The approximate lower bound on K_m is a function of the slew requirements and the number of thruster cycles required to complete the maneuver, as shown in the discussion on slewing maneuvers.

e. Effect of pre-filter gain, K_p

More than any other PWPF modulator parameter, a pre-filter gain is very effective in tailoring the system response. K_p values greater than unity increase any input signal to the PWPF modulator such that the deadband is reached sooner. For a given modulation factor vs. input level curve such as Figure 3.9, the pre-filter gain can be used to selectively boost small error signals $r(t)$ above the deadband so that fine control can be applied without an excessive increase in filter output frequency. Figures 3.11a and 3.11b show the effects of varying K_p on thruster firings on thruster on-time during simulated slew maneuvers. In contrast to increasing K_m , increases in the pre-filter gain up to approximately 10 do not result in excessive thruster firings. This characteristic suggests use of a tunable pre-filter gain which can be optimized over time to tailor the system response. The models discussed in subsequent chapters will utilize this method.

f. Summary of design parameters

Based on the modulator static analysis, the guidelines outlined in Table 3.2 apply to any PWPF modulator design used in on-off thruster applications.

Table 3.2
PWPF Modulator Static Analysis Results

| Parameter | Recommended Setting |
|------------------------|---------------------|
| Modulator Gain, K_m | < 6.0 |
| On-threshold, d | > 0.3 |
| Off-threshold, h | $< 0.8 d$ |
| Pre-filter Gain, K_p | < 15 |

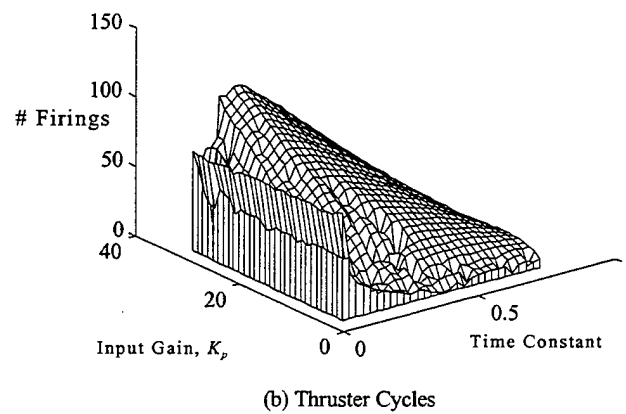
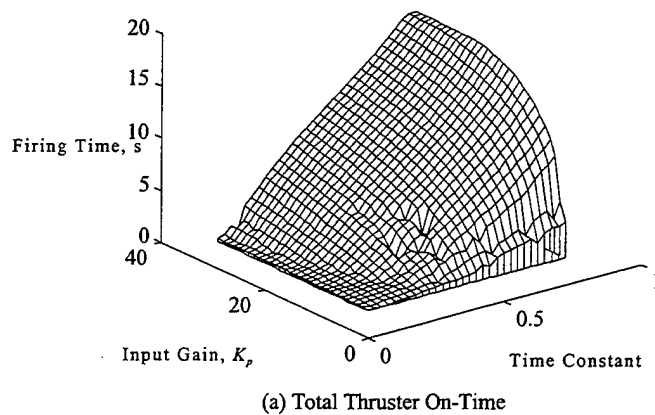


Figure 3.11
Variation in Thruster Frequency and Firing Time, $K_m = 4.5$

2. Dynamic Analysis

The PWPF modulator dynamic response was analyzed in order to quantify the phase lag characteristics for the anticipated range of inputs. Additional information of interest was the relative number of thruster cycles and total on-time. The simulation was conducted by applying unity magnitude sinusoidal inputs ranging from 1 to 150 rad/sec to the PWPF modulator while for various time constant values (0.01 to 0.4). Fixed modulator parameters are shown in Table 3.3 and correspond to the design criteria listed in Table 3.2 above. No pre-filter gain was used.

Table 3.3
PWPF Dynamic Simulation Parameters

| Parameter | Simulation Value |
|-----------|------------------|
| K_m | 4.5 |
| d | 0.45 |
| h | 0.15 |
| U_m | 1.0 |

a. Thruster activity

Figures 3.12 (a) and (b) show the impact of input frequency on thruster activity. For a given time constant, when input frequency increases above a certain value, the firing time is zero. Figure 3.13 shows this trend more distinctly. Figure 3.12 (a) the number of firings is reducing as frequency increases. As opposed to the static analysis, very low values of τ_m resulted in a large total on-times without increasing the number of firings for most cases. The number of firings also increased for low time constants at low frequency. Analysis of Figure 3.12b suggests that the lack of thruster activity at high frequencies is due to loop gain reduction (roll-off). Based on the analysis of thruster activity, a minimum time constant value of 0.1 should be maintained.

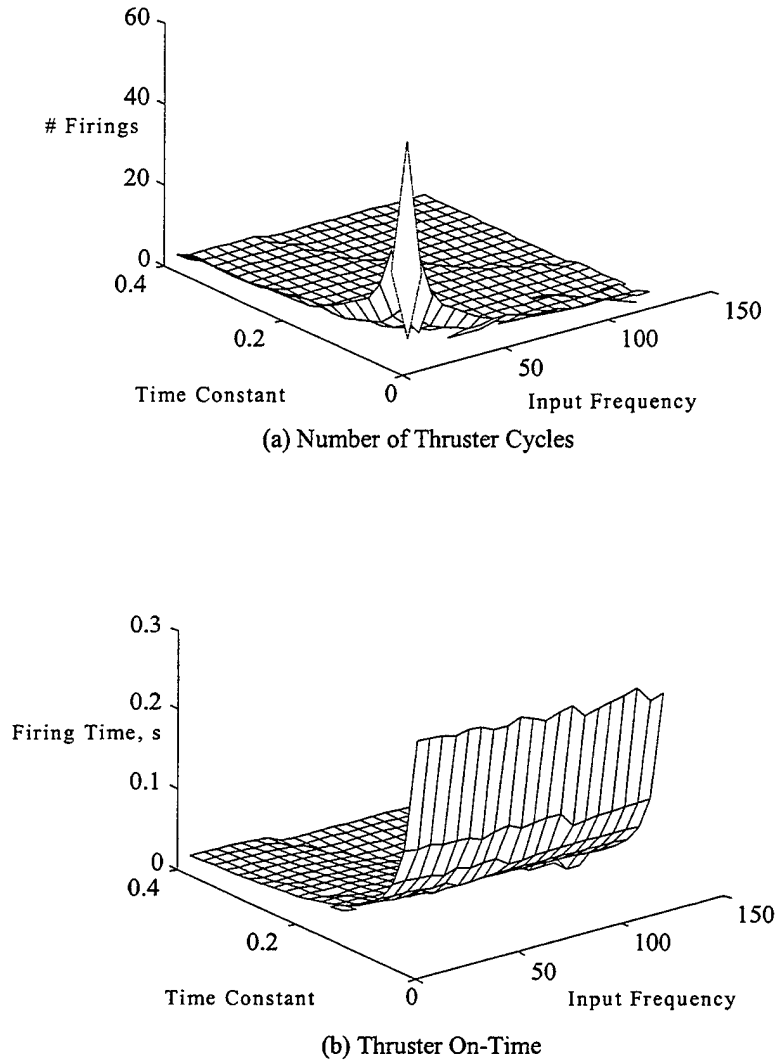


Figure 3.12
Impact of Input Frequency on Thruster Activity

b. Phase lag

Previous research on the PWPF modulator revealed large phase lags which were capable of driving the modulator to an unstable condition, resulting in limit cycle oscillations. The results of the dynamic analysis are shown in Figure 3.13 and reveal some design guidelines. The phase lag, displayed on the vertical axis, is expressed as a fraction of the input phase. For example, an indication of 0 on the vertical scale indicates no phase lag. A value of 1 indicates phase shift on the order of a period.

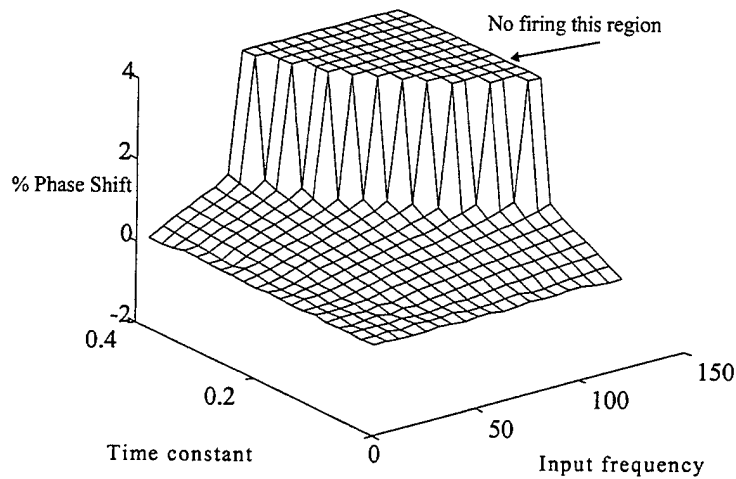


Figure 3.13

Phase Lag Variation with Input Frequency & Modulator Time Constant

Note that for τ_m smaller than 0.2, there is little phase lag for all input frequencies. The plateau shown by a phase of 4 indicates the region of zero modulation factor. In this area, the time constant is too large for the modulator to react to the high frequency input. Note that for τ_m greater than approximately 0.2, the phase lag increases monotonically at low frequency. This characteristic is further reason to maintain τ_m between 0.1 and 0.2 for all applications.

c. *PSD comparison*

The PWPF modulator, while operating in a pseudolinear fashion, can not exactly replicate a linear signal input. The modulator time constant plays a key role in determining the frequency response and the modulator's ability to track an input of a given frequency. The mission impact of this limitation is manifested in the system response when performing slew maneuvers. Figure 3.14 illustrates the input and PWPF modulator output frequency response for varied time constant and frequency settings. Note the energy distribution for small time constant. Increasing the time constant distorts the signal until finally the modulator output is zero. This situation corresponds to an excessive time constant and indicates that the PWPF filter output never reaches the on-threshold. Figure 3.13 was an equivalent representation of this characteristic. These

observations are consistent with the phase lag analysis in that they would suggest a time constant on the order of 0.2 or less.

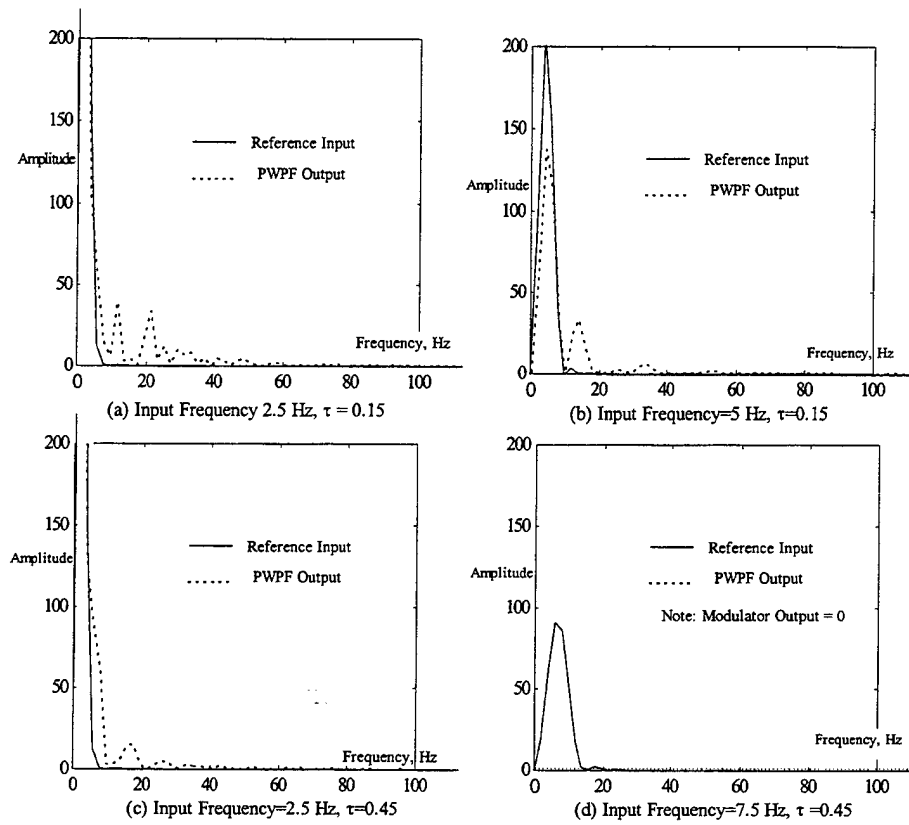


Figure 3.14
Input Signal, Modulator Output Power Spectral Density Comparison

3. Impact of Modulator Parameters on FSS Slewing Performance

Slewing performance of the FSS model with PWPF modulator was analyzed for 10, 20, and 30 degree maneuvers. The slewing commands were issued as unit steps scaled to the appropriate slewing magnitude. Based on the static analysis, K_m , d , h , and U_m were fixed. Pre-filter gains (K_p) from 1 to 30 and modulator time constants (τ_m) from 0.02 to 0.9 were studied. The results of the simulations are illustrated in figures 3.15-3.16 (rigid-body performance) and 3.17-3.18 (flexible mode response). One additional run with a modulator gain of $K_m=1.5$ was conducted and confirmed the relationship between K_m , K_p , and the modulator impact on the slew.

a. *Effect of varying modulator time constant*

As shown in figures 3.15 and 3.17 the modulator time constant directly impacts the rigid body (maximum overshoot and settling time) and flexible responses. That is, for $\tau_m < 0.2$ the rigid body and flexible body responses are nominal. As τ_m increases, interaction between rigid and flexible modes increases, resulting in degraded maneuver performance and increasing residual vibration. For values of 0.8 or greater, the rigid body overshoot is on the order of magnitude of the command and the flexible vibration becomes extreme.

In terms of the lower bound on τ_m , Figure 3.17 shows that for $\tau_m < 0.1$, a dramatic increase in the number of thruster cycles results. However, the total on-time is not increased significantly. These characteristics suggest a series of very short duration pulses might be used to follow a very high frequency command. The design suggestion is to put a lower bound on τ_m at 0.1, allowing judicious tuning to a value less than 0.1 as needed, subject to specific hardware limitations such as thruster minimum on-time.

b. *Effect of modulator gain, K_m*

The rigid-body and flexible time responses were unaffected by modulator gain changes as long as the product of K_p and K_m was larger than approximately 6. At values below 6, the rigid body settling time increased but did so without a commensurate increase in the overshoot or system vibration. The conclusion once again is that the designer should make K_m as low as possible based on the rigid body slewing requirements. If the restrictions on maneuver time are fairly loose, a minimum gain value on the order of 1.5 is acceptable.

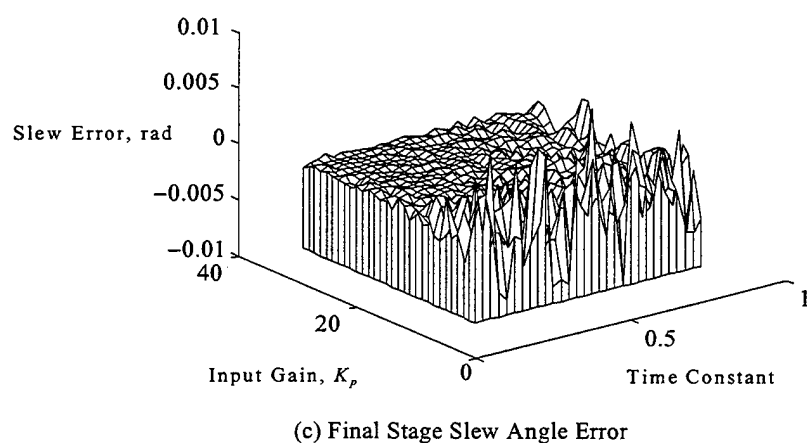
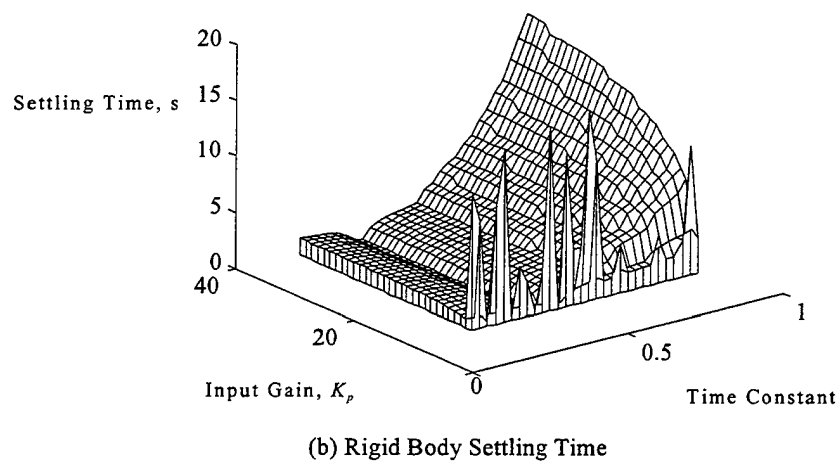
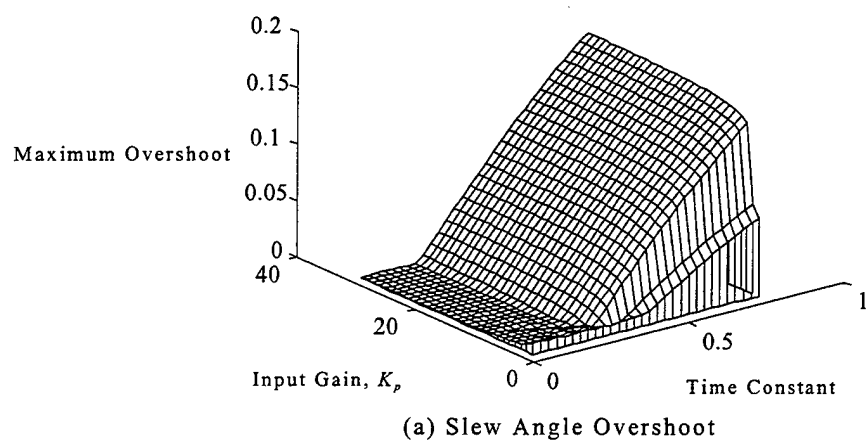
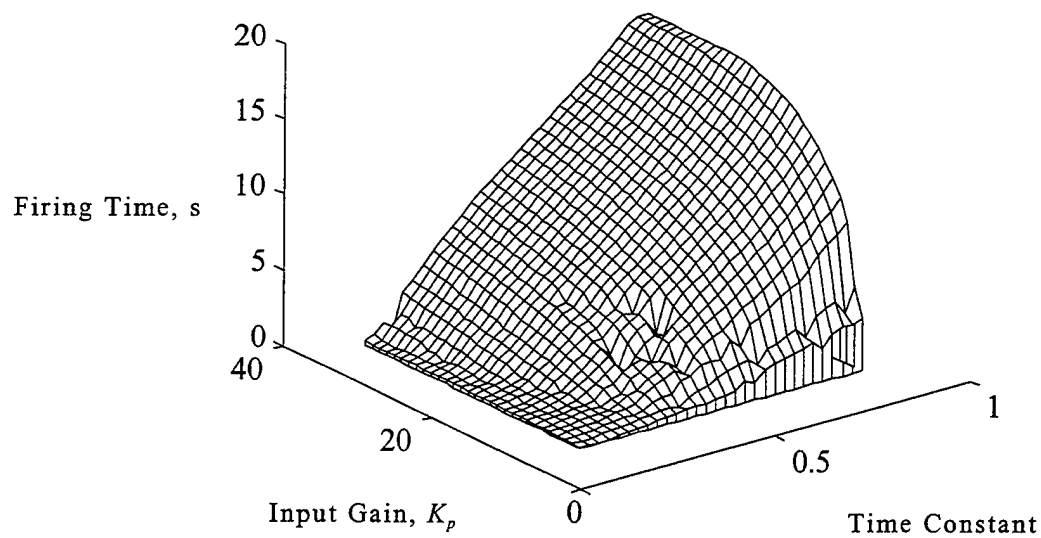
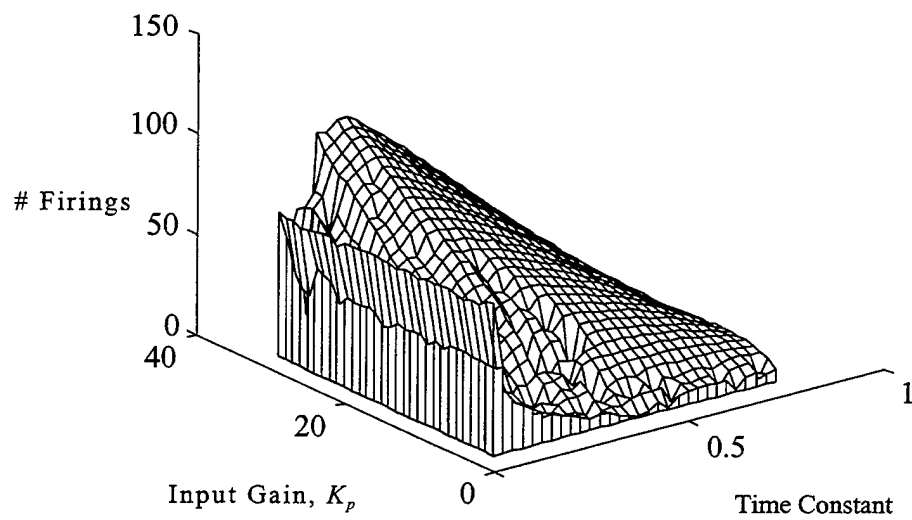


Figure 3.15
Rigid Body Slewing Response, $K_m = 4.5$



(a) Total Thruster On-Time



(b) Total Thruster Cycles

Figure 3.16
Thruster Activity During Slewing Maneuvers, $K_m = 4.5$

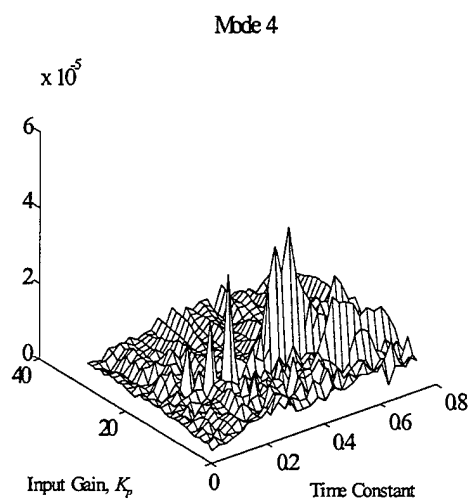
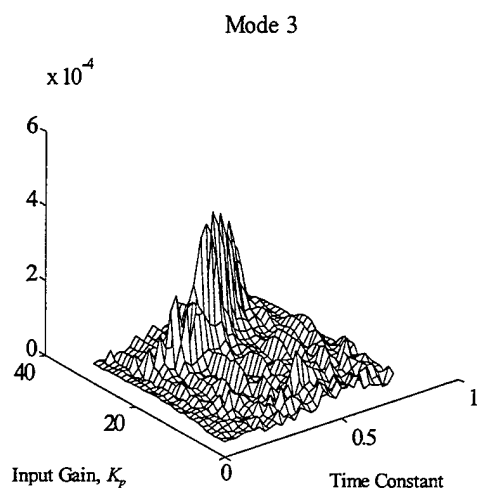
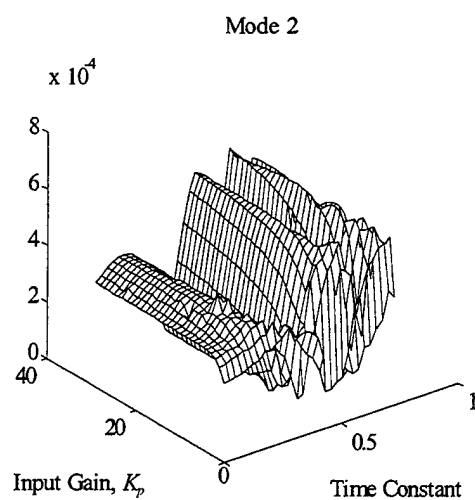
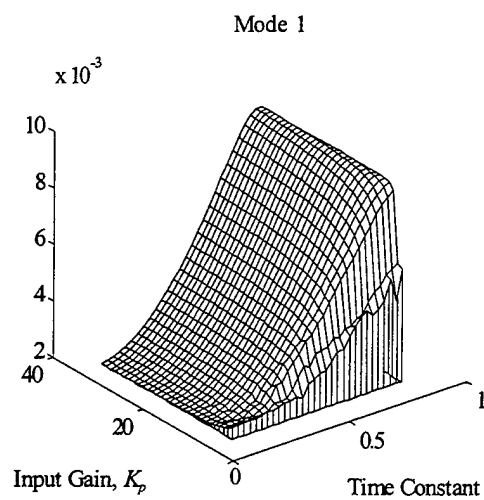


Figure 3.17
Flexible Response to Slewing Maneuvers, $K_m = 4.5$, Modes 1-4

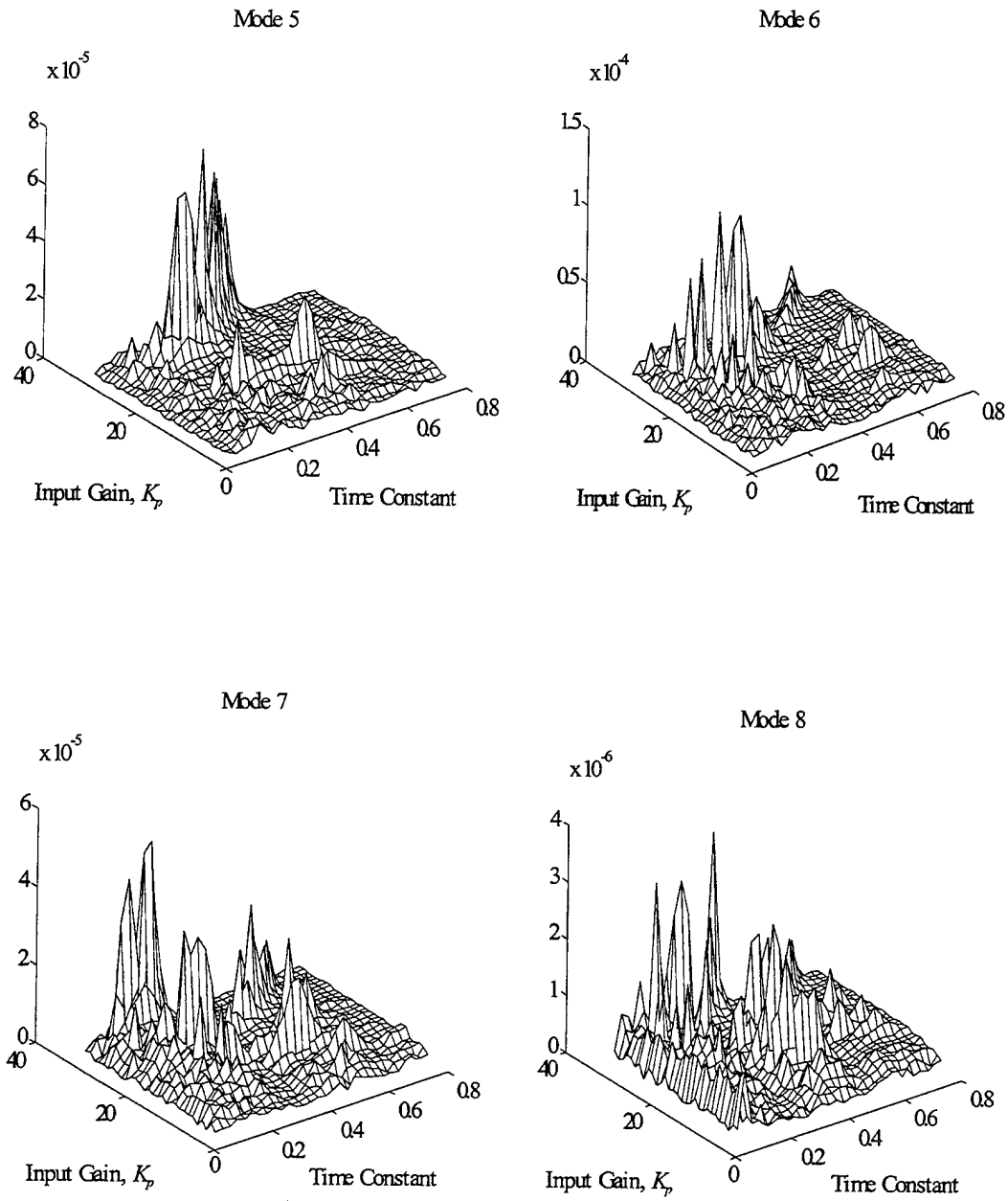


Figure 3.18
Flexible Response to Slewing Maneuvers, $K_m = 4.5$, Modes 5-8

c. Effect of pre-filter gain, K_p

The pre-filter gain, K_p , biases the input signal $r(t)$ to keep it outside the deadband or to limit its maximum value. As discussed in the static analysis, the effective deadband is determined by the product of K_m and K_p , so that varying K_p at a fixed value of K_m tailors the deadband without incurring excessive thruster firings. Note in Figure 3.15 that K_p has little impact on the rigid body performance as long as the product of K_m and K_p is sufficiently large to complete the maneuver. Once K_p is increased to the point where the signal is beyond the deadband, further increases have little effect on the rigid body response. A minimum value of $K_p=2$ is sufficient to guarantee desired performance for most ranges of K_m .

Figures 3.16a (on-time), 3.17 and 3.18 show that large values of pre-filter gain can excite specific flexible modes as a function of modulator time constant. In addition, high pre-filter gain values cause an increase in the number and duration of thruster firings for a given performance. Minimizing the thruster on-time and number of cycles requires imposing a ceiling of 10 on K_p .

4. Design Recommendations

The previous analyses revealed several consistent trends in PWPF characteristics which can aid the designer in selecting appropriate modulator parameters. Few of the parameters are worth tuning and the tunable range is relatively small. However, even small modifications in the pre-filter gain and the time constant can make a significant difference in achieving the desired performance. Table 3.4 summarizes the results and recommended settings for each parameter.

Table 3.4
Summary of PWPF Analyses

| Modulator | Analysis Type | | | Design |
|-----------|---------------|---------|------------|------------------------|
| Parameter | Static | Dynamic | Slew | Recommendation |
| K_m | < 6.0 | N/A | ≥ 1.0 | 1.0-6.0 |
| K_p | < 10 | N/A | ≥ 2.0 | 2.0-10 ⁽¹⁾ |
| τ_m | N/A | 0.1-0.2 | 0.1-0.2 | 0.1-0.2 ⁽²⁾ |
| d | > 0.3 | N/A | N/A | > 0.3 |
| h | $< 0.8 d$ | N/A | N/A | $< 0.8 d$ |

Notes: (1) Tuning or dual-staging recommended.

(2) Judicious use of τ_m below 0.1 is acceptable

Selection of the PWPF parameters for the FSS model and input shaping simulations followed these guidelines. Table 3.5 lists the selected values, including a dual-stage pre-filter gain. The selected configuration resulted in the best system performance (rigid-body and flexible response) prior to application of the input shaper.

Table 3.5
FSS Model PWPF Modulator Configuration

| Parameter | Value |
|-----------|---|
| K_m | 1.25 |
| K_p | 2.0, input $> d/K_m$ 5.0, input $\leq d/K_m$ |
| τ_m | 0.15 |
| d | 0.45 |
| h | 0.15 |

IV. INPUT SHAPING TECHNIQUES

A. BACKGROUND & DESCRIPTION

The goal of input shaping is to provide a command which results in zero residual vibration. The source of these vibrations is arbitrary and may include control commands or disturbance torques. As shown in Figure 4.1, a vibration can be eliminated by applying impulses of appropriate amplitude and phase such that they exactly cancel the mode. For example, the command issued at time t_1 , in this case an impulse, starts a vibration which decays as a function of the modal damping. The second impulse is phased such that it is applied at the vibrating mode's half-period point $(t_2 - t_1) = 1/2 T$. The net vibration following the second impulse is zero.

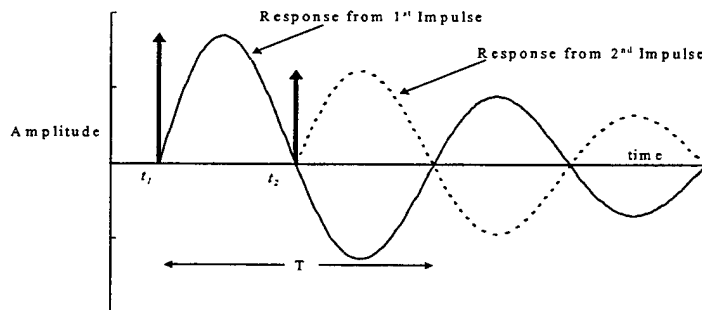


Figure 4.1
Vibration Cancellation due to Input Shaping

Input shaping uses this technique to modify either open- or closed-loop system commands. In general, shaper pulse trains will consist of an initial pulse and some number of additional impulses designed to exactly cancel vibration in a pre-determined number of flexible modes. The initial pulse, which is always applied at the command time, t_1 , has unity magnitude. As such, the *initial* impulse will always be coincident with the command which begins the system vibration. Amplitude and timing of subsequent pulses are determined using the designer's knowledge of plant frequency and damping

characteristics, subject to uncertainty bounds. Referring to Figure 4.1, the initial pulse at t_i is coincident with the desired command. The amplitude of the second pulse is based on the damping of the mode and the half-period point.

Modifying a given command to utilize input shaping is very straight-forward. Given a known set of plant conditions subject to some prescribed uncertainty, the desired command is identified. For example, the command could be an open-loop torque input to a control system or a step position command to a state feedback controller. The time domain system commands are then convolved using linear systems theory. The convolution integral, presented in elementary linear systems texts (Hsu,1995) as

$$y(t) = x(t) * h(t) = \int_{-\infty}^{+\infty} x(\tau) h(t - \tau) d\tau \quad (4.1)$$

shows that the time response $y(t)$ for any linear time-invariant system is the convolution of the arbitrary input $x(t)$ with the impulse response $h(t)$ of the system. The original command is scaled by the magnitude of $h(t)$ and delayed in time by an amount τ .

The net effect of applying an input shaper to an arbitrary command signal is to scale the torque (open-loop) or position (closed-loop) command by the magnitude of the shaper pulse train and to delay each segment of the command by the time τ_i corresponding to the application time of the i 'th impulse in the train. Figure 4.2 shows the result of convolving a four impulse sequence with a step command.

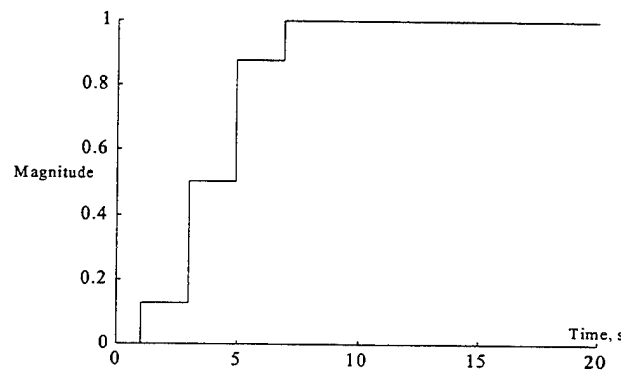


Figure 4.2
Convolution of Impulse Train and Step Command

Note that each "step" of the command corresponds to a specific impulse used to form the modified command. Thus the resulting input commands accomplish the desired maneuver while simultaneously ensuring there is no vibration during the slew. By ensuring that the magnitude of the shaper impulse train is unity, no other modification to the original command is needed.

This chapter will introduce input shapers and describe the method of employing them to single and multiple mode systems. Solutions will be presented for variable-amplitude actuated systems and then extended to constant-amplitude actuator systems.

B. ADVANTAGES OF INPUT SHAPING

Input shaping was first introduced by O.J.M. Smith in the 1950's (Smith, 1958), but the application was oriented toward open-loop systems only. Recently, Singer and Seering showed the technique's applicability to closed-loop systems (Singer and Seering, 1990). Subsequently, extensive research has been performed into the robustness and multiple-mode suitability of input shaping techniques.

Two primary advantages of input shaping techniques are ease of implementation and superior performance over other vibration suppression strategies. Compared to closed-loop time optimal control methods, which require state feedback and are typically quite complex, input shaping is in principle and practice an extremely easy technique to implement. Depending on the degree of plant uncertainty, various shaping techniques are available to ensure minimal vibration during maneuvers. As shown in various research papers, these techniques may be applied to either open- or closed-loop systems with equal success (Pao and Singhose, 1995a; Singhose and Pao, 1996). As a result, input shapers can be seamlessly added to any control system. Modifications to inner loop controllers can be performed without altering the input shaper. From a performance standpoint, sensitive instruments which suffer degraded performance due to vibration can continue operating during maneuvers. Secondly, any residuals which do occur are so small that damping them is greatly simplified.

C. DESCRIPTION OF INPUT SHAPERS

1. Classification by Actuator

There are two general categories associated with input shapers. Early research in the field was slanted toward linear, variable amplitude actuators such as momentum wheels or motors. These actuator types are also referred to as “ideal” or “continuous time” actuators. Due to the linear nature of those formulations, the control shapes could be obtained analytically. Since then the major thrust of the research has been focused on use of unmodulated on-off (or “bang-bang”) controllers, which cannot realize variable amplitude commands. Shapers used with bang-bang control are termed Constant Amplitude Pulse (CAP) shapers. Shapers used with variable amplitude actuators will be termed Variable Amplitude (VA) shapers.

2. Classification by Impulse Train

Several shapers are prevalent in the literature and are named for the method used to arrive at their solutions. Zero Vibration (ZV), Zero Vibration Zero Derivative (ZVD), and Zero Vibration Zero Second Derivative (ZVDD) are all based on driving the impulse response equation for a normalized, decoupled system to zero. These shapers are listed in order of increasing robustness. (Singer & Seering, 1990). The ZV shaper has been shown to lack robustness for even small plant variations. The Extra-Insensitive (EI) shaper is designed to provide even more robustness by allowing the impulse response equation to be non-zero, that is, it allows some small residual vibration. The analyses in this thesis are based on the ZVD and ZVDD shapers because the design goal is zero vibration with acceptable robustness.

D. INPUT SHAPER DESIGN

Input shaping techniques are based on linear systems theory. If a given system is linear it can be described by a set of decoupled modal equations. Vibratory response equations for the individual modes are used to develop the input shaper pulse train.

Additionally, the system damping and frequency are known to within some tolerance value. While not as critical for single-mode cancellation, the tolerances on plant uncertainty become extremely important when attempting to cancel multiple mode vibrations with either constant- or variable-amplitude actuators. Listed below is a general procedure for implementing the input shaper to any open- or closed-loop system.

Step 1) From system equations of motion, determine system natural frequencies and damping information.

Step 2) Find the impulse sequence which will cancel modal vibration: apply the appropriate vibration constraint equations for robustness, actuator type, and time optimality. This sequence is the "shaper command".

Step 3) Identify the desired system command and convolve it with the shaper command to obtain the "shaped system command".

The equations presented below are common in the current literature. Solution of the shaper equations will be discussed first from the standpoint of variable amplitude actuators and then extended to constant amplitude pulse (CAP) actuators. The complete derivations can be found in Singer (1988).

1. Vibration and Robustness Constraint Equations

For a system which utilizes a variable amplitude actuator, the shaped command which eliminates residual vibration in the flexible mode also corresponds to the time optimal command (Pao, 1995b). In this case, the input shaping solution is easily found using vibration constraints alone. The rigid body equations are unnecessary because output levels can be scaled to ensure desired move distances are reached. The number of impulses and shaper type is determined by the robustness and performance requirements.

a. Zero Vibration (ZV) shaper equations

Each mode of an uncoupled, linear vibratory system of arbitrary order is characterized by an impulse response given by

$$y(t) = \left[A \frac{\omega_o}{\sqrt{1.0 - \zeta^2}} e^{-\zeta\omega_o(t-t_o)} \right] \sin\left(\omega_o \sqrt{1.0 - \zeta^2} (t - t_o)\right) \quad (4.2)$$

where A is the amplitude of the impulse, ω_o is the undamped system natural frequency, ζ is the damping ratio for each of the modes, and t_o is the time of the impulse input. The amplitude of the vibration due to a sequence of impulses is given by

$$A_{amp} = \sqrt{\left(\sum_{j=1}^N B_j \cos \phi_j \right)^2 + \left(\sum_{j=1}^N B_j \sin \phi_j \right)^2} \quad (4.3)$$

where

$$\phi_j = \omega \sqrt{(1 - \zeta^2)} t_j \quad \text{and} \quad B_j = \frac{A_j \omega}{\sqrt{(1 - \zeta^2)}} e^{-\zeta\omega(t_N - t_j)}$$

where A_j is the amplitude of the j th impulse, ω is the system natural frequency, t_N is the time of the final impulse, and t_j is the time of the j th impulse. In order to ensure there is zero vibration at the end of the impulse train, each coefficient B_j in Eq. (4.3) must be identically zero, resulting in two simultaneous “Zero Vibration (ZV)” equations

$$\begin{aligned} \sum_{j=1}^N A_j e^{-\zeta\omega(t_N - t_j)} \sin(t_j \omega \sqrt{1 - \zeta^2}) &= 0 \\ \sum_{j=1}^N A_j e^{-\zeta\omega(t_N - t_j)} \cos(t_j \omega \sqrt{1 - \zeta^2}) &= 0 \end{aligned} \quad (4.4)$$

Note that these equations are written in terms of multiple impulses to cancel a single vibrational mode. The shortest pulse train which can cancel a single vibrational mode consists of a two-impulse sequence initiated at $t=0$ with a unity magnitude initial pulse. The amplitude and timing of the second pulse are obtained by solving Eqs. (4.4) simultaneously. Figure 4.3 shows the resulting impulse train. Note that the impulse train amplitudes have been normalized to unity gain. This procedure is necessary to ensure that the shaper does not scale the original command. Otherwise, the input shaper would lose its ability to function with open- and closed-loop systems.

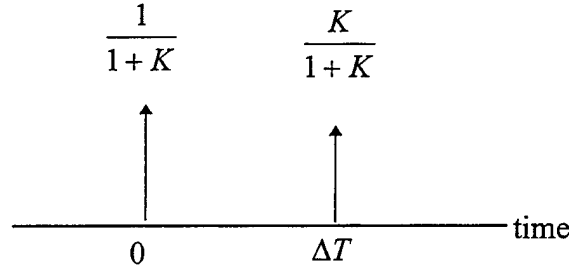


Figure 4.3
Two Impulse ZV Sequence

The pulse train parameters are given by

$$K = e^{-\frac{\zeta\pi}{\sqrt{1-\zeta^2}}} \quad \Delta T = \frac{\pi}{\omega_o \sqrt{1-\zeta^2}} \quad (4.5)$$

b. Zero Vibration Derivative (ZVD) equations

While the ZV shaper provides the shortest impulse trains, it requires very good knowledge of the plant. Singer and Seering (1990) showed that the ZV shaper was robust for only small variations ($\pm 5\%$) in modal frequency. In order to enhance the shaper's robustness, an additional set of constraint equations can be obtained by differentiating Eq. (4.4) with respect to natural frequency, ω :

$$\sum_{j=1}^N A_j t_j e^{-\zeta\omega(t_N-t_j)} \sin(t_j \omega \sqrt{1-\zeta^2}) = 0 \quad (4.6)$$

$$\sum_{j=1}^N A_j t_j e^{-\zeta\omega(t_N-t_j)} \cos(t_j \omega \sqrt{1-\zeta^2}) = 0$$

Satisfying the additional set of constraints requires two additional variables in the form of an additional impulse (A_3 and t_3). Solving the set of equations yields the impulse train illustrated in Figure 4.4 and quantified by Eq. (4.5) above.

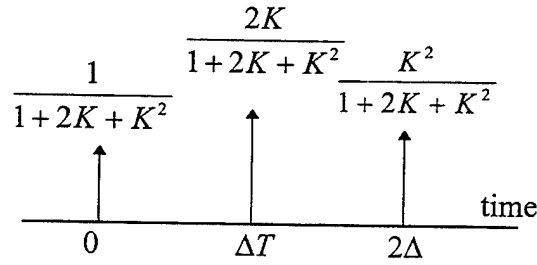


Figure 4.4
ZVD Impulse Train

This technique has been shown to provide robustness for up to $\pm 20\%$ variations in frequency (Singer & Seering, 1990).

c. Zero Vibration Derivative Derivative (ZVDD) shaper equations

If the vibration equations are differentiated once again, the resulting vibration will be zero for a range of frequencies above and below the system natural frequency. The effect is to improve robustness dramatically. As noted by various authors, the ZVDD shaper allows plant uncertainties on the order of $\pm 40\%$ while retaining the zero vibration characteristic (Pao 1996). The additional constraint equations are obtained by differentiating Eq. (4.6) with respect to ω and setting it to zero:

$$\begin{aligned} \sum_{j=1}^N A_j t_j^2 e^{-\zeta\omega(t_N-t_j)} \sin(t_j\omega\sqrt{1-\zeta^2}) &= 0 \\ \sum_{j=1}^N A_j t_j^2 e^{-\zeta\omega(t_N-t_j)} \cos(t_j\omega\sqrt{1-\zeta^2}) &= 0 \end{aligned} \quad (4.7)$$

Once again, the additional set of equations requires two more unknowns, A_4 and t_4 , so that a total of four impulses are needed in the train to cancel the single vibrational mode. Figure 4.5 illustrates the impulse train for a ZVDD shaper.

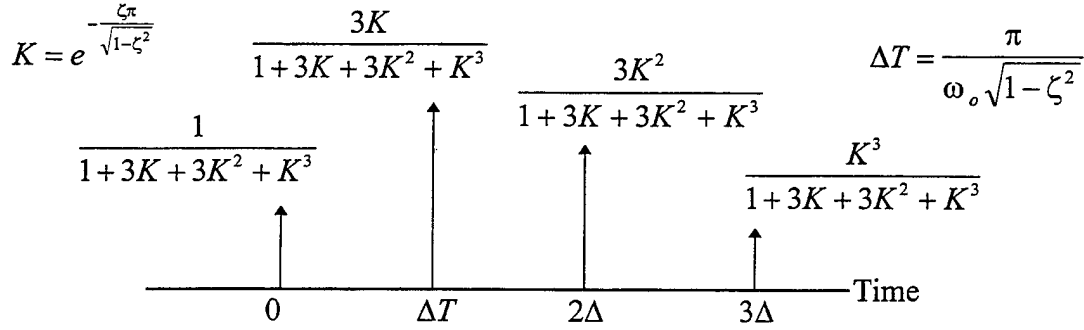


Figure 4.5
ZVDD Shaper Impulse Train

2. Constant Amplitude Pulse (CAP) Shaper Constraints

a. Amplitude constraints

Because CAP actuators may only assume discrete values dependent on the input level, they place different constraints on the system than the variable amplitude case. CAP shapers can be implemented with ZV, ZVD, or ZVDD impulse trains. The net effect of bang-bang control on input shaping comes in the form of an additional constraint equation which must be solved in obtaining the nominal pulse train. A typical impulse profile for a multiple switch bang-bang slew is of the form

$$\begin{bmatrix} A_j \\ t_j \end{bmatrix} = \begin{bmatrix} 1 & -2 & 2 & -2 & 1 \\ 0 & t_2 & t_3 & t_{n-1} & t_n \end{bmatrix} \quad (4.8)$$

Once again, the default initiation time is $t=0$ and each A_j switch toggles the thruster output from positive to negative. For a thruster level of +1, 0, or -1 the impulse amplitudes listed in Eq. (4.8) satisfy the constraint

$$\left| \sum_{j=1}^N A_j \right| = 1 \text{ or } 0 \quad (4.9)$$

In sum, the above constraints dictate the fashion in which residual vibration is canceled for a given actuator type.

b. Rigid-body constraints

An important observation must be made regarding CAP shapers. Because the command amplitude is fixed, the switching profile and command length are functions of the move distance. Accordingly, CAP constraints cannot be applied independently of the rigid body equations like the variable amplitude case was. CAP-shaping a command without regard to the rigid body equations would result in an end-state other than desired. The rigid body equation of motion for FSS slewing maneuvers is given by

$$\ddot{\theta} = \frac{T_t}{I_{zz}} \quad (4.10)$$

where θ is the angular displacement of the rigid body, T_t is the total torque applied, and I_{zz} is the total moment of inertia. For any desired velocity during the profile or for spin-up maneuvers to a desired velocity $\dot{\theta}_d$, integrating Eq. (4.10) yields

$$\dot{\theta}_d = \int_{t_0}^t \frac{T_t}{I_{zz}} dt \quad (4.11)$$

where the applied torque T_t is a function of time and the lower limit of integration can be taken to be identically zero. When a rest-to-rest slew is performed, Eqs. (4.10-4.11) are identically zero at the beginning and end of the maneuver, yielding the constraint equation for slewing distance. Integrating Eq. (4.11), the desired slew distance, θ_d is

$$\theta_d = \int_t \int_t \frac{T_t}{I_{zz}} dt dt \quad (4.12)$$

c. Time-optimality constraint

For a given move distance and system configuration, there exists a unique solution which is time-optimal. However, because the number and timing of the impulses vary with move distance, there are many sub-optimal solutions to the zero-vibration slewing problem (Singhose, et al., 1996). Whereas the variable amplitude actuator case allowed almost trivial solution to the time-optimal problem, the additional constraints imposed by the bang-bang case constitute a set of non-linear optimization criteria which

cannot be solved analytically. For a switching profile such as Eq. (4.7), the time optimality constraint is expressed as

$$\min(t_n) \text{ where } t_n \text{ is the time of the final switch} \quad (4.13)$$

3. Extension to Multiple Modes

In principle, extending the vibration equations to cancel n modes is just a matter of writing the constraint equations for each mode and solving them simultaneously to obtain the pulse train. However, there are some serious complications. The accuracy and ease of the solution depends on the number of modes, the mode ratio (defined as the frequency of each mode as compared to the fundamental), and the move distance in the case of bang-bang commands (Crain, 1996). If the modes are tightly spaced (one extreme) or are widely separated in frequency (the other), the number and timing of impulses vary greatly. Even with variable amplitude shapers, vibration levels in multiple modes are difficult to eliminate. Use of CAP shapers exacerbates the effect due to the high frequency content typical in most bang-bang controllers. As noted in Pao (1995a), the decrease in vibration was no more than 45% for ZVD-CAP shapers and 70% for ZVD-VA shapers. In addition, extreme vibration of higher modes was entrained by the CAP shaper. In general, there is no efficient method to find a multiple mode CAP shaper.

E. SUMMARY

VA shapers and CAP shapers have merits and disadvantages. The VA equations are easy to solve and do not require imposition of the rigid body constraints. However, VA actuators cannot be directly used for on-off thrusters. On the other hand, design of the CAP shaper is much more complicated, even for few flexible modes. Additionally, CAP shapers tend to entrain large values of higher mode vibration. The goal of this research, therefore, is to identify a shaper-controller combination which is as easily designed as the VA shaper but that allows performance associated with on-off thrusters. Pulse-width, pulse-frequency modulation holds the key to this challenge.

V. SHAPED SLEW MANEUVERS USING LINEAR ACTUATORS

A. APPROACH

The effects of VA shapers are determined in order to establish a performance baseline against which maneuvers executed with a PWPF modulator can be compared. A variable amplitude actuator such as a momentum wheel was used to execute the VA shaper commands in the FSS SIMULINK model. In order to demonstrate the shaper's portability, the simulations were performed open-loop and then repeated for a closed loop system with PD control. Open-loop simulations included both step and smoothed torque commands. The open-loop command represents a constant torque input. Initially, only a single flexible mode was included, so that the shaper command generation and application can be more easily understood. The investigation was then repeated for the FSS with eight flexible modes.

B. FSS WITH SINGLE FLEXIBLE MODE

1. Generation of Shaper Commands

The state space equations of motion for the FSS developed in Chapter II apply. Using a single flexible mode, the equations reduce to

$$\begin{bmatrix} I_{zz} & D_1 \\ D_1 & 1 \end{bmatrix} \begin{Bmatrix} \ddot{\theta} \\ \ddot{q}_1 \end{Bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & -2\zeta_1\omega_1 \end{bmatrix} \begin{Bmatrix} \dot{\theta} \\ \dot{q}_1 \end{Bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & -\omega_1^2 \end{bmatrix} \begin{Bmatrix} \theta \\ q_1 \end{Bmatrix} = \begin{bmatrix} T_t \\ 0 \end{bmatrix} \quad (5.1)$$

where the momentum wheel command is given by T_t and the system parameters are as follows:

$$\begin{aligned} D_1 &= -1.6872 \\ \zeta_1 &= 0.004 \\ \omega_1 &= 1.33 \text{ rad/s (0.213 Hz)} \\ I_{zz} &= 10.49 \text{ kg-m}^2 \end{aligned}$$

a. ZVD Shaper commands

Using Eqs. (4.3 and 4.5) to target zero residual vibration in the single flexible mode, the ZVD shaper equations are solved using $N = 3$:

$$\sum_{j=1}^N A_j e^{-\zeta\omega(t_N-t_j)} \sin(t_j\omega\sqrt{1-\zeta^2}) = 0$$

$$\sum_{j=1}^N A_j e^{-\zeta\omega(t_N-t_j)} \cos(t_j\omega\sqrt{1-\zeta^2}) = 0$$

$$\sum_{j=1}^N A_j t_j e^{-\zeta\omega(t_N-t_j)} \sin(t_j\omega\sqrt{1-\zeta^2}) = 0$$

$$\sum_{j=1}^N A_j t_j e^{-\zeta\omega(t_N-t_j)} \cos(t_j\omega\sqrt{1-\zeta^2}) = 0$$

Recall that the first impulse has unity magnitude at time $t_1 = 0$. As a result there are four equations with four unknowns A_2 , t_2 , A_3 , and t_3 . The resulting ZVD shaper sequence is comprised of three impulses, normalized to unity as shown in Figure 5.1:

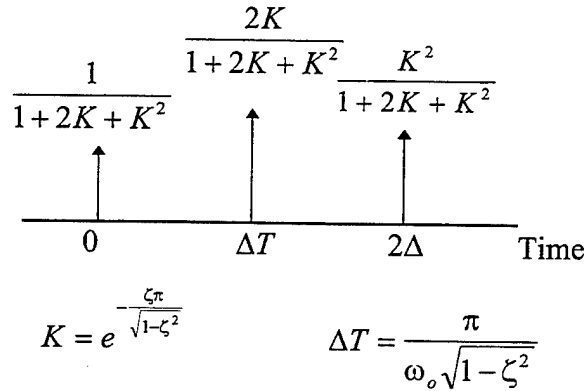


Figure 5.1
Single-mode ZVD Shaper for FSS

b. ZVDD Shaper commands

Similarly, equations (4.3, 4.5, and 4.6) are used to find the ZVDD input shaper command. In this case, a total of four impulses is required to solve the shaper constraint equations. Again, the pulse train is unity normalized. The ZVDD shaper pulse train is shown in Figure 5.2:

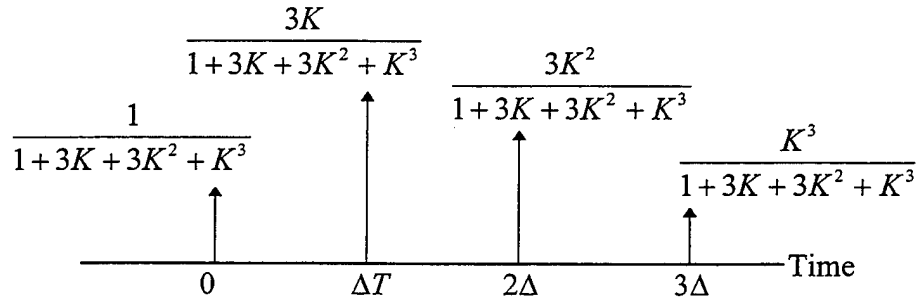


Figure 5.2
Single Mode ZVDD Shaper Pulse Train

c. Comparison of shaped and unshaped commands

Convolving the shaper pulse trains with a step position command yields the shaped command profiles for applying to the open- or closed-loop systems. Figure 5.3 illustrates the difference in command profiles.

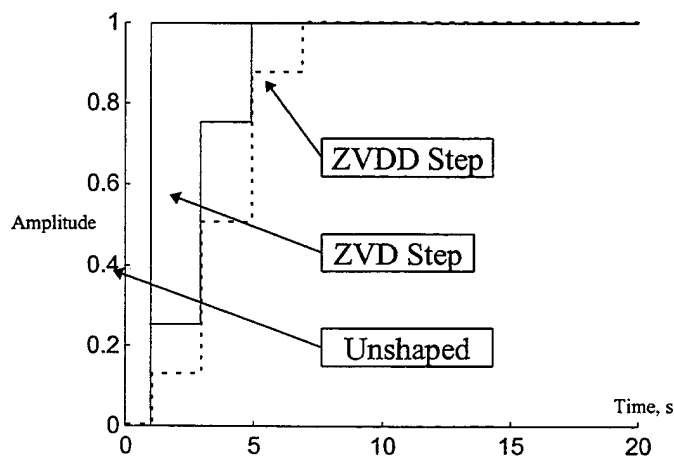


Figure 5.3
Comparison of ZVD, ZVDD, and Unshaped Command Profiles

2. Open Loop System Response

Two cases are studied. In the first simulation, the open-loop response to a step command is obtained as a worst-case condition. In the second, a smoothed torque command is used as a performance baseline corresponding to an ideal linear actuator. The SIMULINK block diagram of the open-loop system is shown in Figure 5.4:

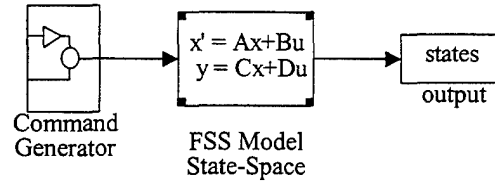


Figure 5.4
Smoothed Torque Command (open-loop) Simulation

a. Response to step command

Inputting an open-loop step command results in a steady state angular velocity and large values of vibration in the flexible mode. Figure 5.5a shows the difference between the shaped and unshaped step inputs. The first mode has a non-zero mean displacement due to the static input. Applying a ZVD- or ZVDD-shaped command completely eliminates the modal vibration. After the vibration is canceled, the flexible appendage has a static displacement which is proportional to the slew rate. Figure 5.5b illustrates the impact of the input shaper. Using a linear actuator allows exact cancellation of the vibration, since there are no unwanted frequencies included in the actuator output.

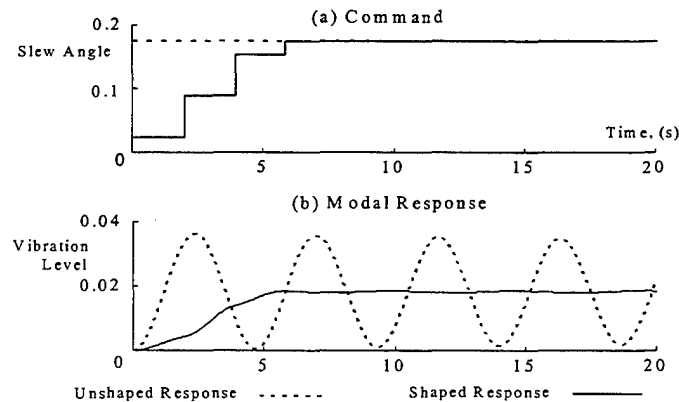


Figure 5.5
Open-loop Step Command Response

b. Response to smoothed torque command

Applying a step command without having any closed loop control is illustrative but not realistic. Therefore, response to a more typical smoothed torque command is analyzed to provide a comparison to the shaper control. This type of command is also known as a “pre-computed torque” profile. In contrast to a step input, which has infinite jerk, the smoothed command applies the acceleration slowly and causes less modal vibration. The smoothed command used in this simulation is generated by using a fifth-order polynomial curve to join a zero command level to a unity command level in a user-defined time period.

Figure 5.6a shows the smoothed input command and the rigid body response. Figure 5.6b shows the flexible mode response to the smoothed command. Note that there is still some residual vibration, despite the smooth start and finish on the torque command. In fact, the “rounded edges” on the command serve to avoid exciting high frequency modes but do not eliminate low frequency vibration. Thus, smooth torque profiles are more effective for multiple mode systems than they are for single mode systems.

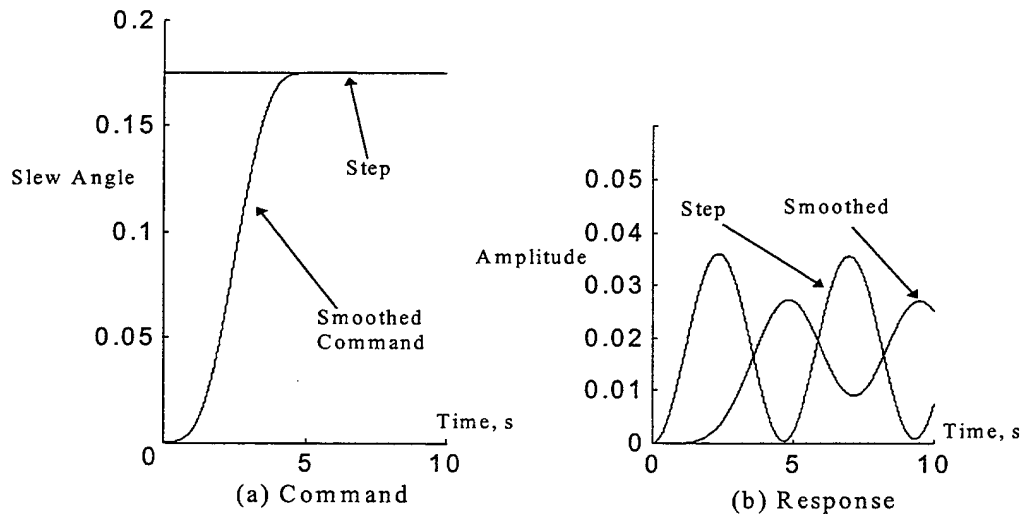


Figure 5.6
Smoothed Input Command, System Response

3. Closed Loop Slew Maneuvers

Adding a PD controller as illustrated in Figure 5.7 allows closed-loop attitude control for the FSS. Simulations for slewing maneuvers of 10, 20, and 30 degrees using PD gains of $k_p = -10$ and $k_d = -20$ were analyzed. Command profiles included unshaped step, ZVD, and ZVDD shaped inputs as depicted in Figure 5.3 previously.

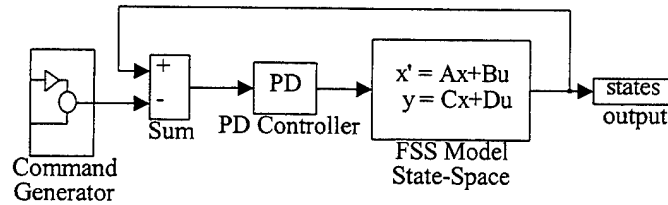


Figure 5.7
Closed loop Slewing System using PD controller

The rigid-body and flexible mode responses to each command type are presented in figures 5.8a and 5.8b. Note that the unshaped slew is characterized by considerable modal vibration. Both the ZVD and ZVDD shapers eliminate the vibration almost immediately, but the rigid body settling times differ by the additional length of the ZVDD shaper pulse train.

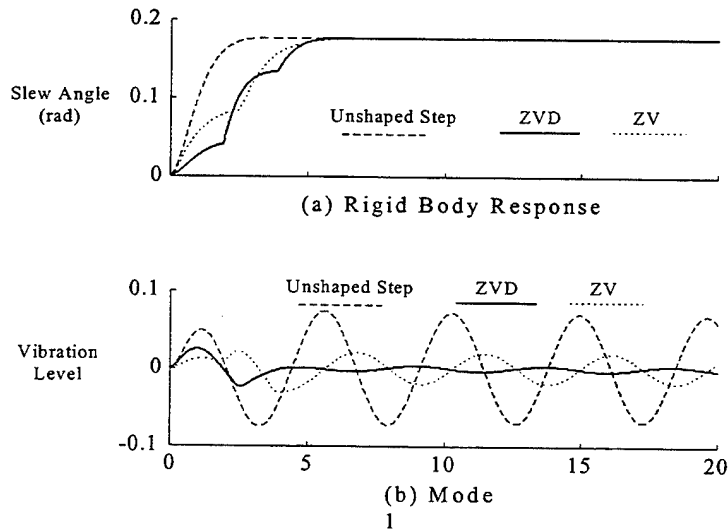
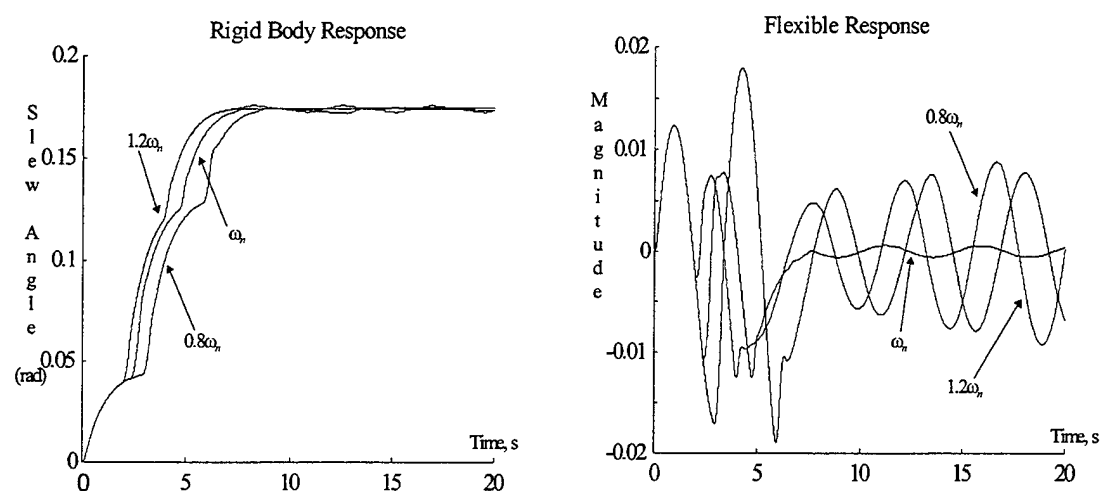
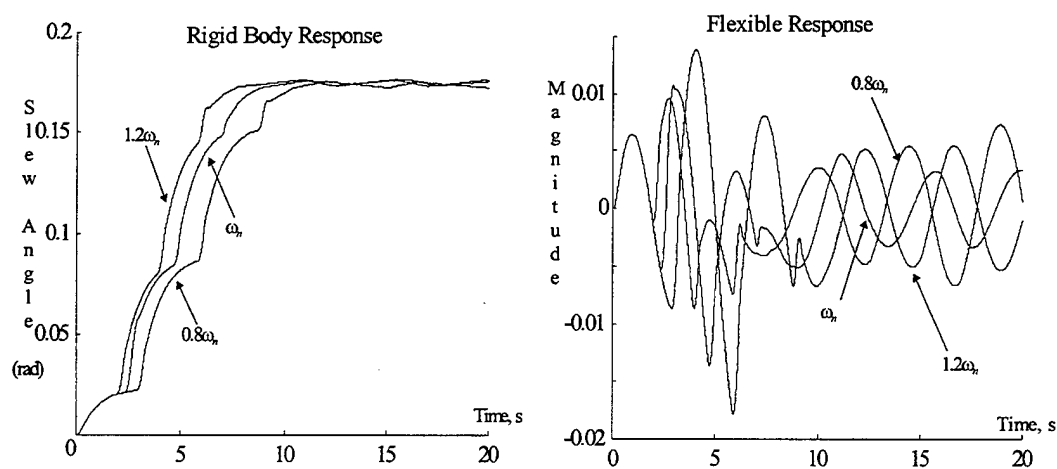


Figure 5.8
Closed Loop Responses to Shaped & Unshaped Commands

These simulations show that input shaping is highly effective for single mode operation when the plant is known. Even in the presence of frequency uncertainty, the ZVD and ZVDD controllers can perform well. Figure 5.9 compares robustness of ZVD and ZVDD shapers for frequency uncertainties up to 20%. The decision to use the ZVDD shaper over the ZVD shaper is based on a trade-off between maneuver time and robustness. Only if the plant is well-known or non-varying will the ZVD shaper provide better response time while ensuring residual vibrations are eliminated.



(a) Single Mode ZVD Shaper



(b) Single Mode ZVDD Shaper

Figure 5.9
Effects of $\pm 20\%$ Plant Uncertainty on Residual Vibration

C. FSS WITH MULTIPLE FLEXIBLE MODES

Including multiple modes in the FSS model results in the state space formulation developed in Chapter II. The damping ratio for all modes was set at 0.004 for these simulations in order to give a worst case vibration environment. Control torques were applied only to the rigid body. Table 5.1 lists the system natural frequencies for convenience. Up to five modes will be targeted for cancellation in the simulations.

Table 5.1
FSS System Modal Frequencies

| Mode | Frequency | | Period (sec) |
|------|-----------|-------|-----------------|
| | (rad/sec) | (Hz) | |
| 1 | 1.34 | 0.213 | 4.69 |
| 2 | 3.16 | 0.504 | 1.98 |
| 3 | 15.23 | 2.42 | 0.446 |
| 4 | 26.71 | 4.25 | 0.235 |
| 5 | 52.94 | 8.43 | 0.119 |
| 6 | 77.31 | 12.30 | 0.081 |
| 7 | 104.2 | 16.58 | 0.060 |
| 8 | 132.1 | 21.02 | 0.047 |

1. Target Mode Selection

Determining the number and mix of modes to cancel is a design issue which is not easily resolved *a priori*. From a practical standpoint, targeting higher modes requires increasingly narrow pulse widths which may test the response time of an actuator or the minimum impulse bit of a thruster. Table 5.1 shows how tightly the periods are spaced at the FSS model's higher frequencies. For systems with even lower fundamental frequencies than the FSS, the crowding occurs progressively sooner.

A more insidious complication in applying input shaping to multiple mode systems is the relationship between the flexible modes. In some cases, cancellation of a lower mode may actually excite the higher mode(s). Pao et. al. (1995) found that as additional modes are targeted, more and more high frequency vibrations are entrained.

Note that the mode ratios r of the FSS model, defined as the ratio of a given frequency to the fundamental, are all greater than ten, with the exception of the second mode. This will prove to be an obstacle to minimizing vibration. The simulations included in this section will demonstrate these difficulties and suggest a strategy for achieving the best results.

In order to show the effectiveness of the various input shapers and the impact of the target mode selection, several cases were selected for this investigation. Table 5.2 lists the targeted modes and the shaper type used in each case. ZVD shapers were less successful than the ZVDD shapers at targeting more than three modes. Only the ZVDD shaper simulations are included for these cases.

Table 5.2
Multiple Mode Input Shaper Targets

| Number of Modes | Targeted Modes | Shaper Type |
|-----------------|----------------|-------------|
| 2 | 1, 3 | ZVD |
| 3 | 1, 2, 3 | ZVD |
| 4 | 1, 2, 3, 4 | ZVDD |
| 5 | 1, 2, 3, 4, 5 | ZVDD |

2. Generation of Shaper Commands

Suppose we desire to target n -modes for cancellation. We have two options for constructing the shaper. The vibration and constraint equations may be solved directly, which results in the shortest possible impulse sequence. This has been shown to be the time-optimal response in the presence of flexibility (Pao, 1995). A simpler approach is to identify the n pulse trains needed to cancel the individual modes and convolve them. While simpler, convolved shaper commands consist of more impulses than direct shaper commands. On the other hand, Crain (1996) showed that as the number of targeted modes and mode ratio increase beyond three the efficiency of convolved shapers approaches that of direct shapers. Due to the large number of modes included in the FSS and large mode ratios, convolved shaper commands are used in this research. Shaper commands to cancel multiple modes are generated as follows:

- Step 1. Identify the modal frequencies targeted for cancellation.
- Step 2. Using the vibration and robustness equations, find the ZVD or ZVDD shaper pulse trains required to cancel the individual modes.
- Step 3. Convolve pulse trains for each mode to obtain the shaper command.
- Step 4. Convolve the shaper command with the desired system command to obtain the zero-vibration system command.

a. ZVD shaper commands

Equations (4.3) and (4.5) are used to obtain the ZVD shaper pulse trains for each of the targeted modes. The general form of the ZVD shaper for each mode is

$$\text{Mode } i: \begin{bmatrix} A_j \\ t_j \end{bmatrix} = \frac{1}{X_D} \begin{bmatrix} 1 & 2K & K^2 \\ 0 & \Delta T & 2\Delta T \end{bmatrix} \quad (5.2)$$

where $K = e^{-\left(\frac{\zeta\pi}{\sqrt{1-\zeta^2}}\right)}$ and $\Delta T = \frac{\pi}{\omega_o \sqrt{1-\zeta^2}}$

The sequences are unity normalized by $X_D = 1 + 2K + K^2$ resulting in the ZVD shaper impulses for modes 1-5 of the FSS:

$$\begin{aligned} \text{Mode 1: } \begin{bmatrix} A_j \\ t_j \end{bmatrix} &= \begin{bmatrix} 0.2532 & 0.5 & 0.2469 \\ 0 & 1.9563 & 3.913 \end{bmatrix} \\ \text{Mode 2: } \begin{bmatrix} A_j \\ t_j \end{bmatrix} &= \begin{bmatrix} 0.2532 & 0.5 & 0.2469 \\ 0 & 0.8273 & 1.655 \end{bmatrix} \\ \text{Mode 3: } \begin{bmatrix} A_j \\ t_j \end{bmatrix} &= \begin{bmatrix} 0.2532 & 0.5 & 0.2469 \\ 0 & 0.1719 & 0.3437 \end{bmatrix} \\ \text{Mode 4: } \begin{bmatrix} A_j \\ t_j \end{bmatrix} &= \begin{bmatrix} 0.2532 & 0.5 & 0.2469 \\ 0 & 0.0980 & 0.1960 \end{bmatrix} \\ \text{Mode 5: } \begin{bmatrix} A_j \\ t_j \end{bmatrix} &= \begin{bmatrix} 0.2532 & 0.5 & 0.2469 \\ 0 & 0.0495 & 0.0989 \end{bmatrix} \end{aligned} \quad (5.3)$$

b. ZVDD shaper commands

Similarly, equations (4.3), (4.5), and (4.6) are used to find the pulse trains for the ZVDD shaper. The resulting four impulse sequence for each mode is given by

$$\text{Mode } i: \begin{bmatrix} A_j \\ t_j \end{bmatrix} = \frac{1}{X_{DD}} \begin{bmatrix} 1 & 3K & 3K^2 & K^3 \\ 0 & \Delta T & 2\Delta T & 3\Delta T \end{bmatrix} \quad (5.4)$$

where K and ΔT are defined in 6.1 and the sequence is unity normalized by

$$X_{DD} = 1 + 3K + 3K^2 + K^3$$

The resulting ZVDD pulse trains for modes 1-5 of the FSS are

$$\begin{aligned} \text{Mode 1: } \begin{bmatrix} A_j \\ t_j \end{bmatrix} &= \begin{bmatrix} 0.1274 & 0.3773 & 0.3726 & 0.1227 \\ 0 & 1.9563 & 3.9127 & 5.8690 \end{bmatrix} \\ \text{Mode 2: } \begin{bmatrix} A_j \\ t_j \end{bmatrix} &= \begin{bmatrix} 0.1274 & 0.3773 & 0.3726 & 0.1227 \\ 0 & 0.8273 & 1.6547 & 2.8420 \end{bmatrix} \\ \text{Mode 3: } \begin{bmatrix} A_j \\ t_j \end{bmatrix} &= \begin{bmatrix} 0.1274 & 0.3773 & 0.3726 & 0.1227 \\ 0 & 0.1719 & 0.3437 & 0.5156 \end{bmatrix} \\ \text{Mode 4: } \begin{bmatrix} A_j \\ t_j \end{bmatrix} &= \begin{bmatrix} 0.1274 & 0.3773 & 0.3726 & 0.1227 \\ 0 & 0.0980 & 0.1960 & 0.2940 \end{bmatrix} \\ \text{Mode 5: } \begin{bmatrix} A_j \\ t_j \end{bmatrix} &= \begin{bmatrix} 0.1274 & 0.3773 & 0.3726 & 0.1227 \\ 0 & 0.0495 & 0.0989 & 0.1484 \end{bmatrix} \end{aligned} \quad (5.5)$$

c. Comparison of shaped and unshaped commands

Once the impulse trains are defined, the user may choose how many modes to target and convolve only those of interest together with the system command. The length of the resulting input shaper sequence is 3" for the ZVD shaper and 4" for the ZVDD shaper. Figure 5.10 illustrates a generic two-mode ZVD shaper (nine impulses) which could be convolved with a system command to complete the desired maneuver with zero vibration in modes one and two.

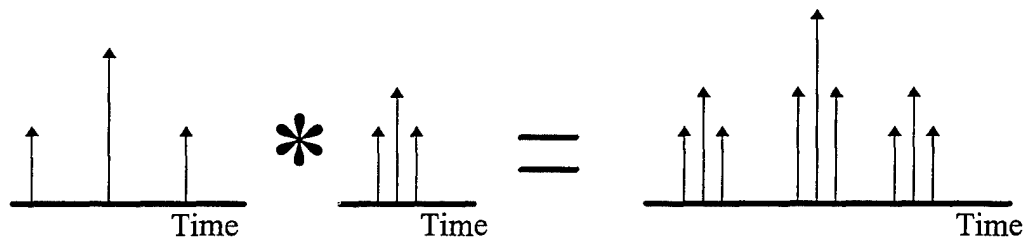


Figure 5.10
Two-mode ZVD Shaper Pulse Train

Clearly, targeting more than three or four modes results in a tightly spaced command pulse train. In the limit as the impulses $\rightarrow 0$, the resulting shaped system command is identical to the smoothed torque profile discussed above. For illustration, Figure 5.11 shows the shaped system commands for a three mode ZVD shaper, a five mode ZVDD shaper, and an unshaped step system command. The five mode ZVDD shaper, with 1024 elements, has the character of the smoothed torque profile but retains enough impulse character to target the residual vibration of lower modes.

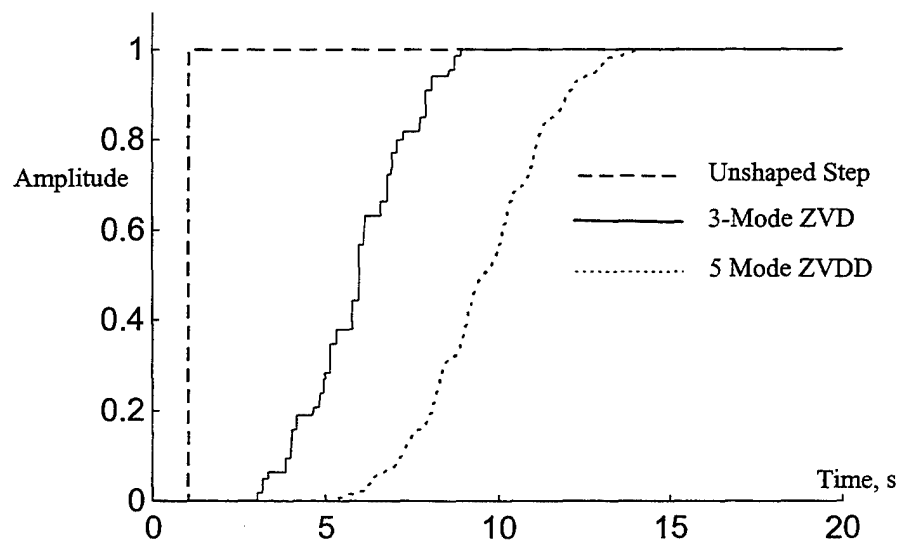


Figure 5.11
Comparison of Shaped and unshaped commands

3. Open Loop System Response

Open loop simulations involving multiple modes of the FSS were conducted in order to make single-to-multiple mode and linear-PWPF control comparisons. Both shaped and unshaped step commands were applied to the system. Smoothed commands were then applied as a more typical open loop type of command. Note that the open loop commands were not full rest-to-rest commands.

a. Response to step command

As shown in Figure 5.12, the 4-Mode ZVDD shaper performs admirably in canceling all four of the targeted modes. Essentially no additional vibration is entrained in the higher modes. In sum, these results were anticipated. The PWPF modulated simulation will reveal telling differences in the robustness and vibration suppression capability between linear and PWPF modulated cases.

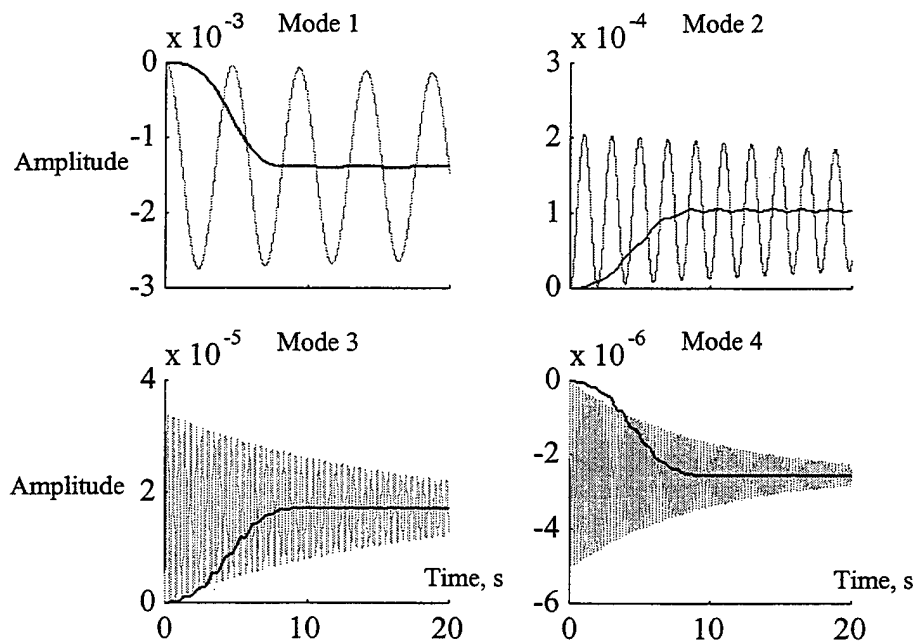


Figure 5.12
Flexible Response to Step Commands (gray = unshaped, black = 4ZVDD)

b. Response to smoothed torque command

The smoothed torque command is a very realistic command found in most control system applications. For the rest-to-rest slew case, the open loop command would be both a smooth ramp up in torque followed by a ramp back down. For the purposes of this analysis, the end state vibration is less important than the vibration during the slew. Therefore, only the first "half" of the computed torque profile is used.

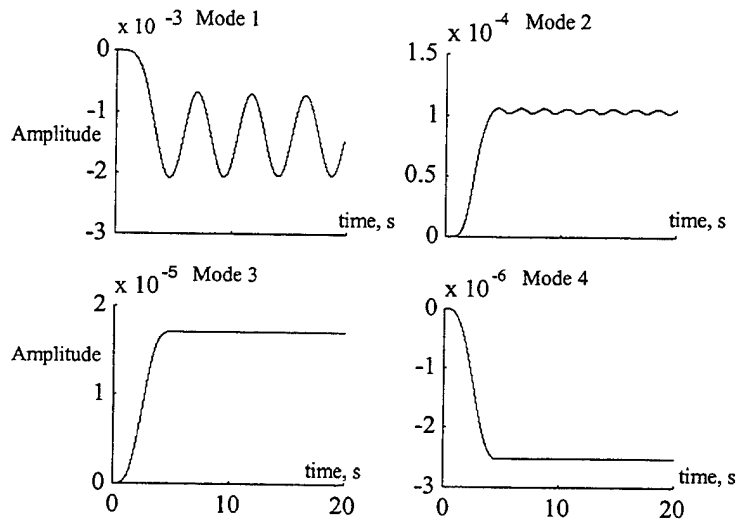


Figure 5.13
Flexible Response to Smoothed Torque Command

4. Closed Loop Slew Maneuvers

Using the same PD controller as used in the single flexible mode example, closed loop slewing maneuvers were conducted to show the effects of input shaping on the residual vibrations of multiple modes. 10-, 20-, and 30-degree slews were performed to investigate the effects of move distance.

D. DISCUSSION

Input shaping can be used very effectively to cancel single flexible modes or widely spaced multiple modes with linear actuators. Because there is no need to use the rigid body, bang-bang, and time optimality constraint equations, solutions for shaper

pulse trains are very easy to obtain and implement. The technique is portable, integrating seamlessly with equal effectiveness for open-loop or closed-loop applications.

However, there are diminishing returns on the performance gain as the number of targeted modes increases. Tightly spaced impulse trains can entrain vibration in the higher frequency modes and defeat the original objective. The number and selection of targeted modes and the shaper type play crucial roles in this process. Detailed knowledge of these effects *a priori* is unlikely.

One solution is to utilize a staged approach with a combination of vibration suppression schemes. Input shaping works well to eliminate the lower mode vibrations, but a different approach should be reserved for the higher modes. For example, a high-bandwidth active controller such as a piezo-electric (PZT) velocity feedback system could be used without adding significant complexity or mass to the system. However, any negative impact of one control type on the other must be identified. For example, a PZT controller has the potential to destabilize the system in certain implementations. This and other options are ripe for further research efforts.

Much of the research into input shaping is directed toward bang-bang control applications. However, that area is much more complex and less fruitful than shaping for linear actuators. If a method existed to use the VA input shapers on a bang-bang system, the best of both worlds should be realized. The following chapter will investigate applicability of the PWPF modulator to realize this goal.

VI. SHAPED SLEW MANEUVERS USING PWPF MODULATOR

A. MOTIVATION

Previous researchers have used variable amplitude input shaping only with linear actuators. This study represents the first application of variable amplitude input shaping to a modulated thruster control system. The results will show that this technique is a simple and effective means of minimizing residual modal vibration in thruster controlled systems. Prior to demonstrating the new approach two common shaping techniques, variable amplitude (VA) and constant amplitude pulse (CAP) shapers are reviewed.

While variable amplitude actuators can produce less vibration in flexible spacecraft than bang-bang actuators, they can become saturated during high-torque maneuvers. Linear thrusters could provide the required torque levels but have problems with valve contamination and leakage. Since most spacecraft must rely on thruster systems for attitude control during station-keeping maneuvers, the main focus of current research has been on optimizing maneuvers with bang-bang actuators.

CAP input shapers allow some degree of vibration cancellation in on-off thruster actuated systems with multiple flexible modes. Singhose, Pao and Seering (1996) reported cancellations ranging from 20-60% using ZVD-CAP shapers. However, there are two major drawbacks for CAP shapers. First, CAP shapers can entrain severe higher mode vibrations. Modal excitations in excess of 800% have been reported (Pao and Singhose 1995). Second, obtaining CAP shaper pulse trains requires nonlinear optimization in the presence of a complicated solution space (Crain, 1996). In light of these limitations, a more efficient method for accomplishing vibration reduction in on-off thruster actuated systems is needed.

The integration of PWPF modulation with a VA shaper offers a solution to this dilemma. The PWPF modulator itself has two primary advantages: its pseudo-linear operation and its capacity for real-time parameter tuning. Unshaped PWPF modulated thruster control has been shown to excite fewer modal vibrations than bang-bang controllers (McClelland, 1994). A variable amplitude input shaper can take advantage of the PWPF pseudo-linear operation. There are several major benefits from this integration.

From a practical standpoint, multiple-mode ZVDD VA shaper pulse trains are easily obtained. High values of vibration suppression should be available without incurring excessive maneuver time penalties. The controller should be robust to frequency variations and can be modified real-time in the presence of varying plant conditions. Finally, if operation of the PWPF modulator is sufficiently linear, high-frequency vibration entrained during the slew maneuvers should be less than that reported for the CAP shaped commands.

B. INTRODUCTION TO SIMULATIONS

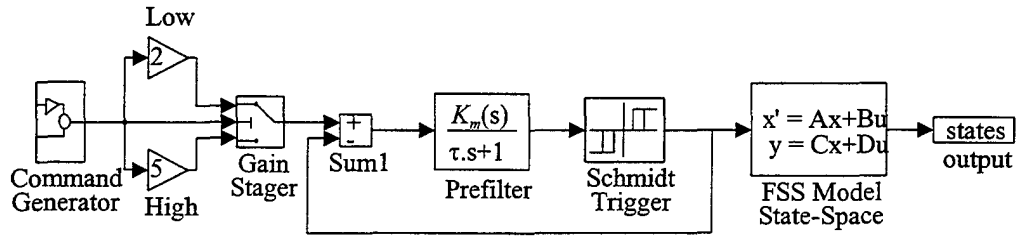
1. Shaper Selection

The proposed implementation uses the VA shaper. Therefore, the sequences already determined for the single mode or multiple mode ZVD or ZVDD shapers will suffice. The number of modes to target will be chosen *a priori* and then analyzed to determine the suitability of the choice. Based on the discussions in Chapter V, no more than five modes of the FSS will be targeted. The shaped system commands, therefore, remain as they were in Chapter V.

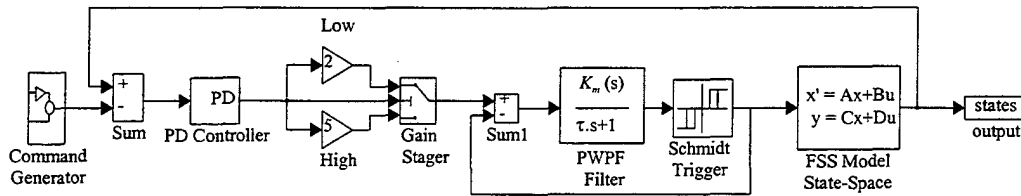
The choice to use a ZVD or ZVDD shaper rests primarily in the tradeoff between robustness and command length. Comparisons made in Chapter V showed these clearly. In the simulations which follow, both types of shapers will be analyzed and any significant differences will be discussed.

2. Simulation Models

Simulations for both single and multiple mode models were conducted using the open- and closed-loop diagrams shown in figures 6.1a and 6.1b. For closed loop simulations, the PD controller was varied to obtain a desirable rigid body response independent from modal vibration levels.



(a) Open Loop PWPF Simulation



(b) Closed Loop PWPF Simulation

Figure 6.1
Shaped PWPF Simulation models

C. FSS WITH SINGLE FLEXIBLE MODE

Single mode simulations were performed in order to provide a comparison of the PWPF modulated response to the ideal, linear system and to isolate the impact of varying the PWPF modulator parameters on vibration cancellation. Performance identical to the ideal system indicates almost perfectly linear operation of the modulator. Modulator deviation from linearity is indicated by degraded vibration cancellation under the same simulation conditions. Variations in PWPF parameters were investigated during the single mode analysis to determine if there is a preferred modulator configuration. Consistent with the previous simulations, both step and smoothed torque commands were applied. The shaped commands remain the same as in Chapter V.

1. Open Loop System Response

a. System performance

Figure 6.2 shows the flexible response to an unshaped step, a ZVD shaped step and a ZVDD shaped step command. Without tuning the PWPF modulator parameters, there is an immediate improvement of 50-60% in the vibration level.

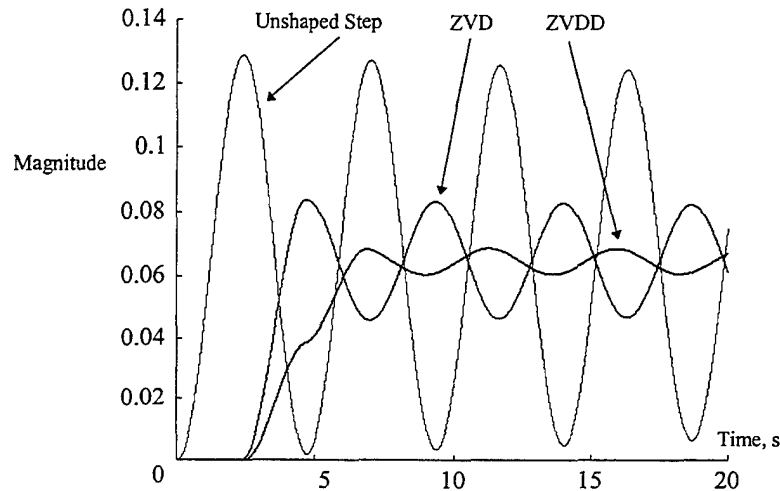


Figure 6.2
Flexible Mode Response, Shaped, single-mode PWPF

In contrast to the ideal system, the single mode is not completely eliminated. This indicates that the frequency content of PWPF output does not exactly match the desired profile. Recalling Figure 3.5, a comparison of the PWPF modulator output power spectrum and the commanded spectrum explains the increase in residual vibration. The PWPF power spectrum includes additional frequency content which is not directed at eliminating the residual vibration. Nevertheless, a large decrease in vibration can be realized, giving a hint of the shapers' inherent robustness. Once the shaper has been chosen, the vibration reduction might be improved by tuning the modulator parameters.

In Chapter V, a smoothed torque command was used to show a more typical open loop command profile. Recall that the smoothed command had the greatest impact on high frequency modes as compared to the fundamental. For the single mode

case, some reduction in vibration was realized, but the performance did not match the shaped input results. In this analysis, the smoothed torque profile is now realized with the PWPF modulator. As expected, the shaper is more effective at reducing the single, low-frequency mode than is the smoothed command. Figure 6.3 shows the modal responses due to the three command types.

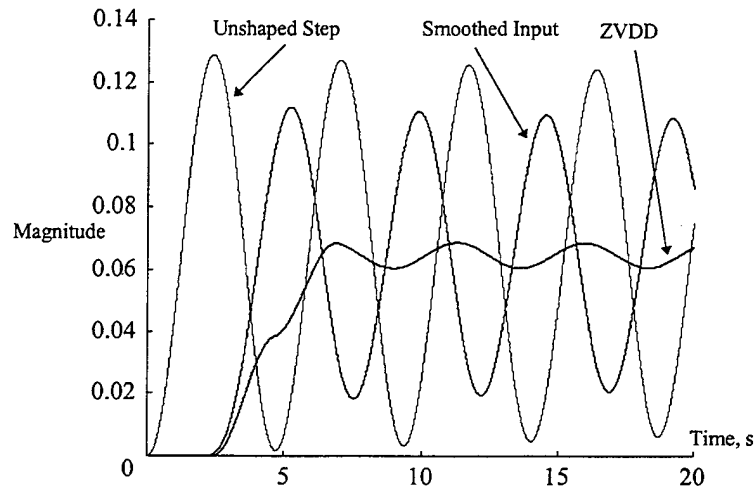


Figure 6.3
Comparison of Smoothed and shaped step command responses

b. Impact of tuning PWPF parameters

The vibration reduction obtained in the last two simulations was performed with a dual-stage, tuned PWPF modulator. Prior to performing any tuning of the modulator, however, the VA shaper was able to reduce vibration to varying degrees. Consistent with the design criteria discussed in Chapter III, the modulator gain, pre-filter gain, and time constant were then varied against each shaper configuration to determine if superior vibration cancellation could be attained. Time constant variations within the design range yielded effectively no change in the vibratory response. This is consistent with the PWPF analysis in Chapter III. Figure 6.4 shows the general trends in tuning the K_p and K_m , respectively, from the high-end of the design range to the minimum values. Note that for K_p fixed, as K_m is varied from $6 \rightarrow 4 \rightarrow 2$, the vibration response varies considerably. For K_m fixed, slight variations from the nominal value of $K_p = 2$ result in considerable change in the vibratory response. An additional case for a modulator gain

outside the design range ($K_m=20$) was included for completeness. While the increase in vibration for $K_m=20$ may not be excessive, the static modulator analysis showed an excessive number of thruster cycles. In summary, nominal values can be identified for both gain parameters.

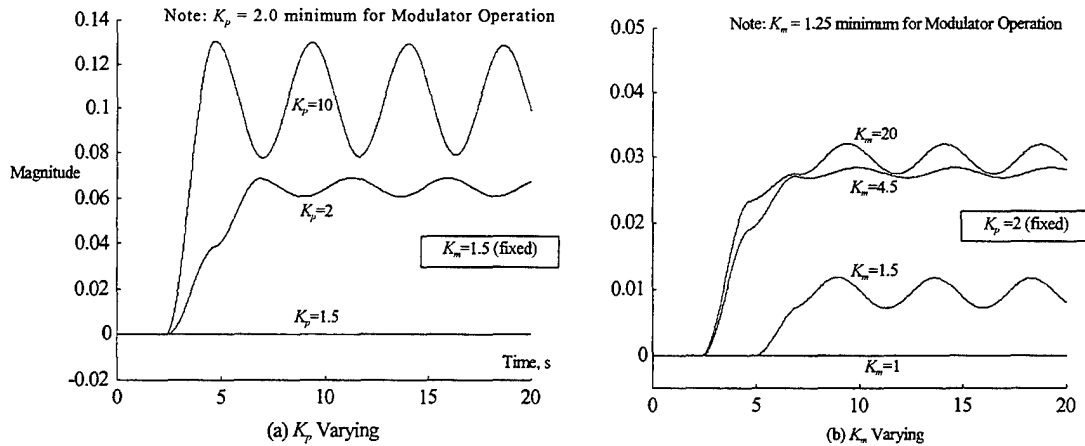


Figure 6.4
Effect of varying K_m and K_p on Vibration Cancellation

Two important observations can be made. First, the fact that minor variations in the modulator parameters can make significant changes in the response suggests that the modulator's tunable range, though seemingly narrow, is sufficient to cover various plant configurations. Second, there appears to be a specific pre-filter/modulator gain combination which results in minimum vibration. Finally, dual-staging the pre-filter gain can have a dramatic impact on the response. Figure 6.4(b) suggests that a gain value of $K_m = 4.5$ is preferred over the recommended design value ($K_m=1.5$). However, addition of the dual stage pre-filter gain with values as indicated in Figure 6.5 clearly shows the advantage of the lower modulator gain. Of note is that the threshold setting for dual staging is based on the error signal effective deadband d/K_m , where d is the Schmidt trigger on-threshold.

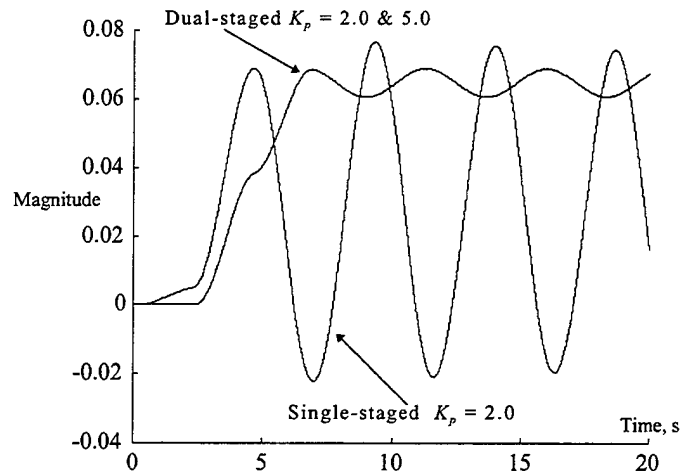


Figure 6.5
Impact of Dual-Stage Pre-filter Gain

c. ZVD vs. ZVDD shapers

A single mode in the ideal system can be canceled with any of the ZV, ZVD, or ZVDD shapers. Introduction of the PWPF modulator makes the degree of vibration reduction a function of shaper type. The additional frequency content resulting from the PWPF modulation process is, in effect, a robustness test. As shown in Figure 6.6, the ZV, ZVD, and ZVDD shapers have varying degrees of success in eliminating the single mode. ZVDD, being the most robust, achieves the best results.

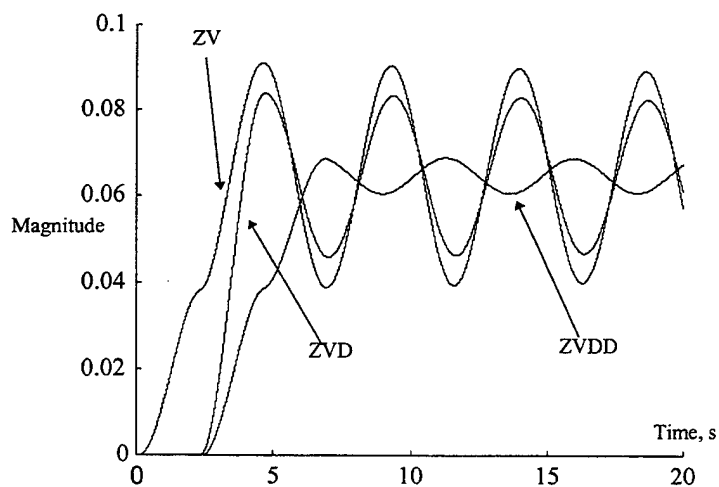


Figure 6.6
Comparison of ZV, ZVD, and ZVDD shapers

2. Closed Loop Slew Maneuvers

a. System response

Ten, twenty, and thirty degree slewing maneuvers were simulated using the closed loop model illustrated in Figure 6.1b. The flexible responses shown in Figure 6.7 indicate that the vibration level caused by an unshaped input command is a function of the move distance. This conclusion is consistent with current research (Singhose, *et al*, 1995) into CAP actuators.

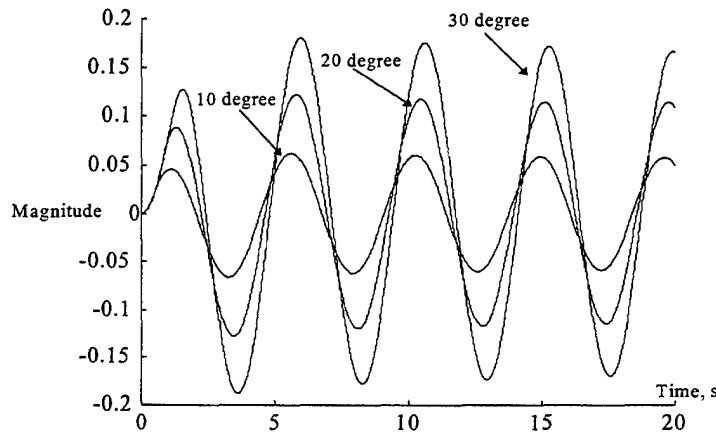


Figure 6.7
Flexible Mode Response to Unshaped Commands

In each case, significant vibration reduction can be realized by using the PWPF modulator to execute a variable amplitude ZVDD shaped command. The reduced vibration can be obtained with little penalty in slewing performance. Figure 6.8 illustrates the additional time required and the vibration reduction for each slewing case.

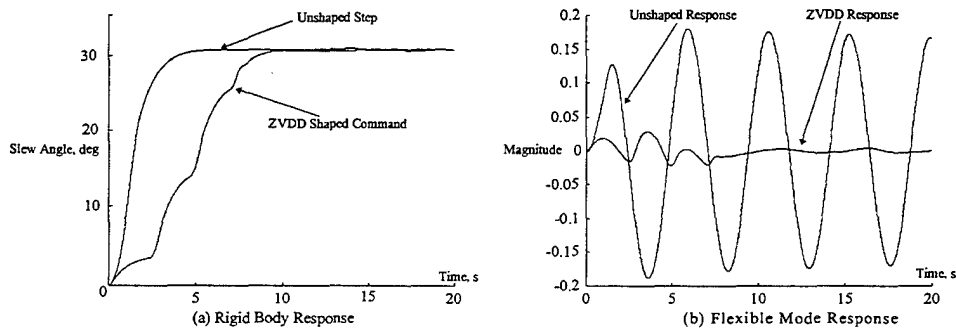


Figure 6.8
System Response, Shaped vs. Unshaped Slews

b. Impact of tuning PWPF parameters

Consistent with the observations made in the PWPF analysis and open loop analysis, the PWPF parameters are chosen to tailor the rigid-body slewing performance while minimizing vibration. The pre-filter gain is dual-staged as described in Chapter III so that the error signal going to the modulator can be brought out of the deadband when desired. The modulator time constant is selected for the rigid body response and the gains are directed toward minimizing vibration. In summary, the modulator selections used for the open loop case do not differ significantly from the closed loop case.

3. Shaper Selection

As expected, the both the ZVD and ZVDD shapers provide considerable vibration elimination. The rigid-body performance costs associated with the ZVDD shaper is minimal, but the additional robustness to frequency variations make it the shaper of choice for this application. The remaining simulations utilized ZVDD shapers to target various flexible modes for cancellation.

D. FSS WITH MULTIPLE FLEXIBLE MODES

Up to this point, simplified simulations have been used to understand the input shaping methodology and to identify the most effective configuration of PWPF modulator and shaping device. This section will report on effectiveness of the FSS with eight flexible modes, a PWPF modulator and multi-mode ZVDD input shaper to perform closed-loop slewing maneuvers. This combination of shaper and actuator has not been previously researched. Based on the observations in the earlier simulations, there is significant potential for this configuration to achieve excellent results.

1. Description of Simulations

The simulations included here validate the use of VA shaper commands to a bang-bang actuator controlled by a PWPF modulator and show an improvement in higher mode excitations than reported in the current literature. Several cases will be analyzed using

ZVDD shapers to cancel the first three, four, and five modes. The PWPF modulator settings remain unchanged from the single mode example, but the PD controller gains were adjusted as necessary to achieve desired rigid-body responses.

2. System Response to Slew Maneuvers

Figure 6.9 shows the unshaped and ZVDD shaped step commands to be executed by the PWPF-controlled system.

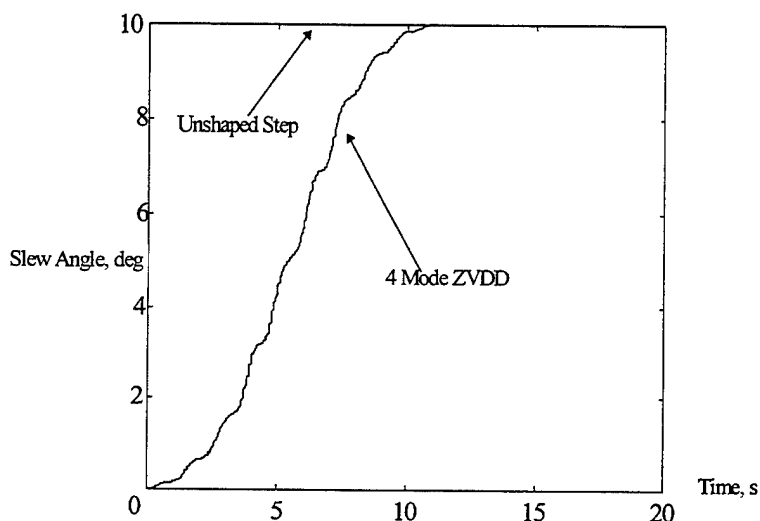


Figure 6.9
Commands and resulting Modal Excitations

Figure 6.10 shows the lower-mode excitations resulting from a ten degree slew maneuver. With modal damping ratios of 0.004, the lower-mode flexible response is essentially undamped for the duration of the unshaped step command simulation. Using a four-mode ZVDD shaper with the PWPF modulator results in excellent cancellation of the targeted modes. Reductions in modal excitations of up to 96% are achieved in the first two modes and approximately 50-60% in the third mode.

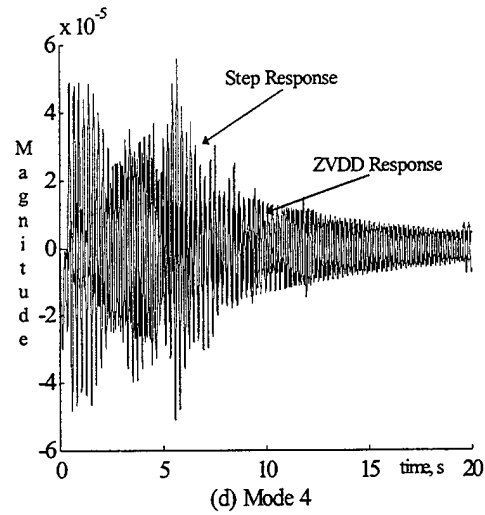
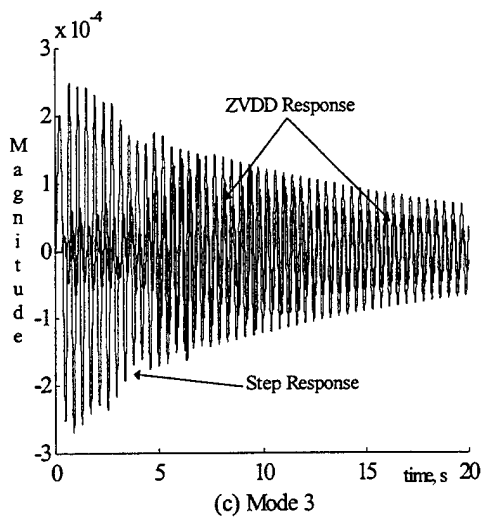
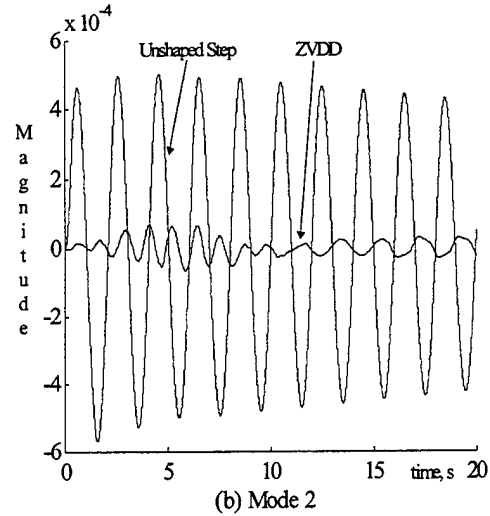
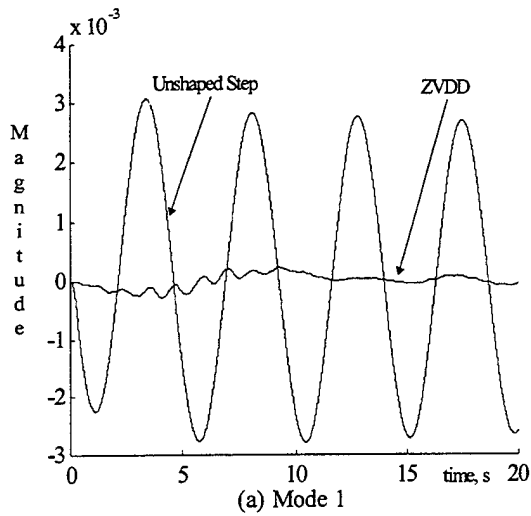


Figure 6.10
Flexible Response, Modes 1-4

Modes three and higher become increasingly difficult to eliminate. Figure 6.11 compares the high frequency responses due to an unshaped step and the four-mode ZVDD shaped commands.

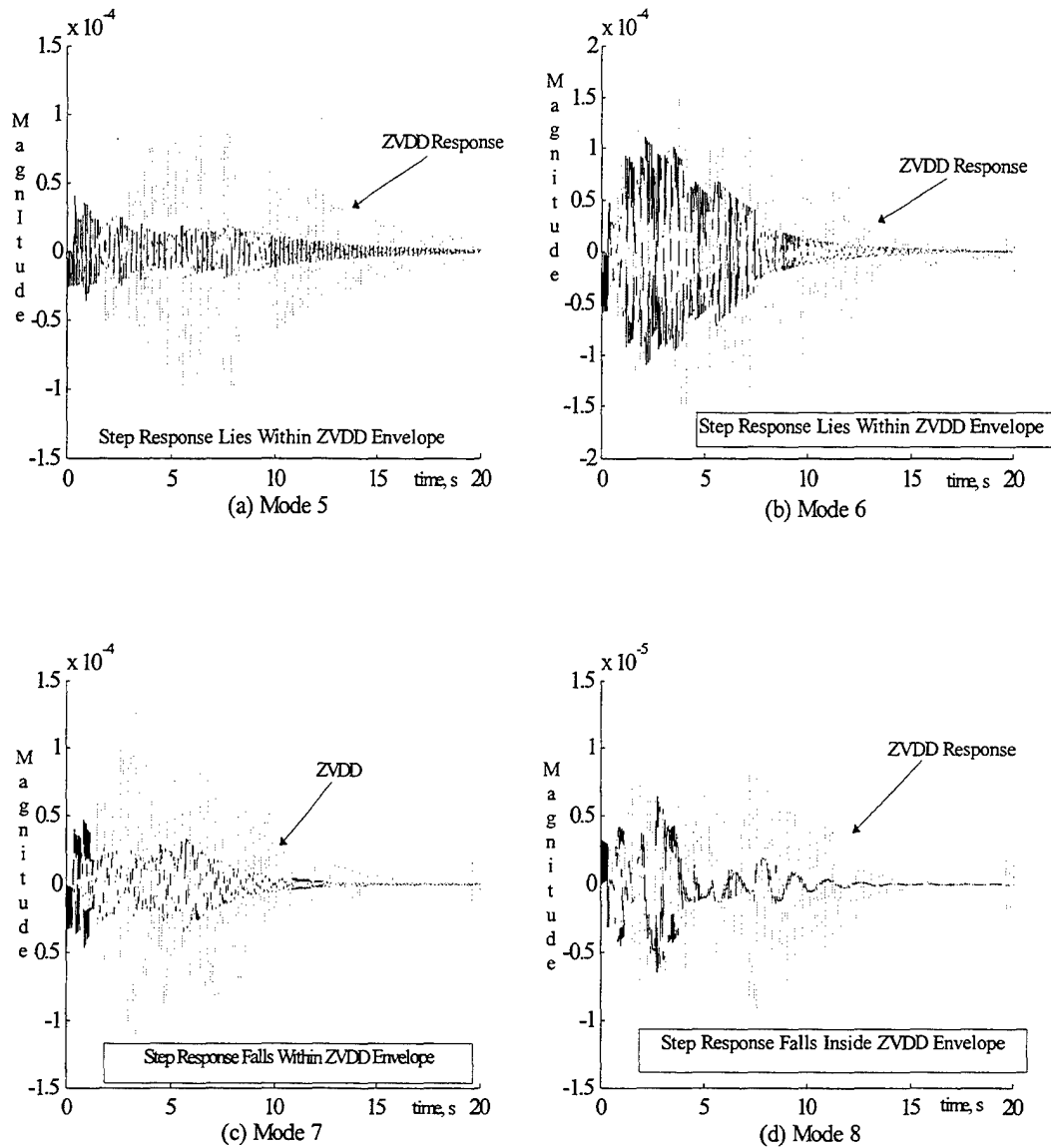


Figure 6.11
High Mode Excitation, ZVDD Shaped Slews

These results are consistent with current research in that there is vibration entrainment in the higher modes. However, the key improvement is that there is no *additional* vibration entrained beyond that generated by an unshaped command. Use of PWPF to execute VA shaper commands substantially improves performance over the CAP shapers reported in current research (Pao, et. al.) without significantly degrading the slewing performance.

3. Robustness

a. Approach

ZVDD shapers have been shown in current literature to be more robust than ZV or ZVD shapers. Pao (1996) reported ZVDD robustness to frequency variations between 30 and 40%. ZVD shapers were shown to possess a minimum of 20% frequency robustness. ZV shapers, while providing the fastest performance, were the least robust, with frequency tolerances on the order of 5% or less.

In this thesis, several shaper/PWPF modulator combinations were analyzed to assess robustness and determine if a ZVD shaper is sufficiently robust to justify its use over a ZVDD shaper when implemented with the PWPF modulator. Final stage error and flexible mode average absolute displacement were obtained using frequency variations from $0.2\omega_n$ to $2.0\omega_n$ and damping variations of 0.1ζ to 2.0ζ . Rigid body and flexible mode responses for frequency variations of $\pm 20\%$ were recorded. The ZVDD shapers were considerably more robust to frequency uncertainty. The results from the ZVDD case are reported here and comparison is made between single- and multi-mode ZVDD shaper robustness characteristics.

b. Results

Figures 6.12 and 6.13 show rigid body and flexible mode responses for frequency variations of up to 20% from nominal. Using average absolute displacement as the performance metric, Figure 6.14 shows the robustness of a four-mode ZVDD shaper implemented with the PWPF modulator. Figure 6.15 shows the robustness of a single-mode ZVDD shaper. Ideal performance is characterized by an average displacement of zero. From these plots, three general conclusions can be made regarding the frequency and damping robustness of the ZVDD shapers.

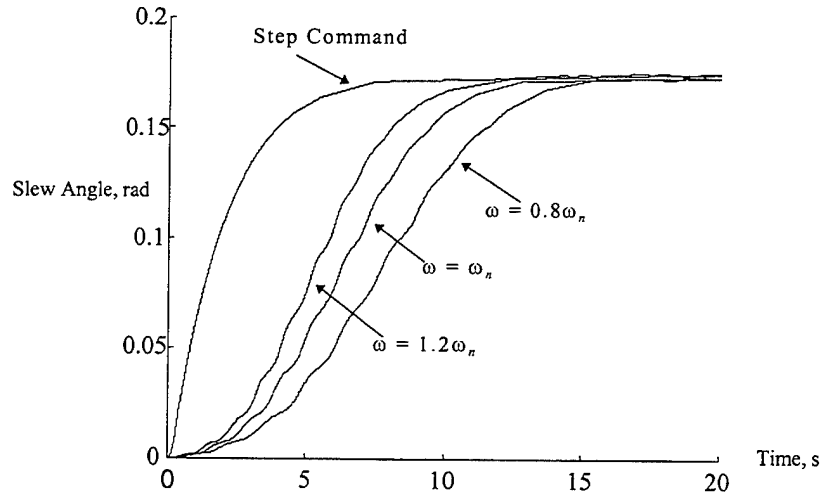


Figure 6.12
Rigid Body Response to 20% Frequency Uncertainty

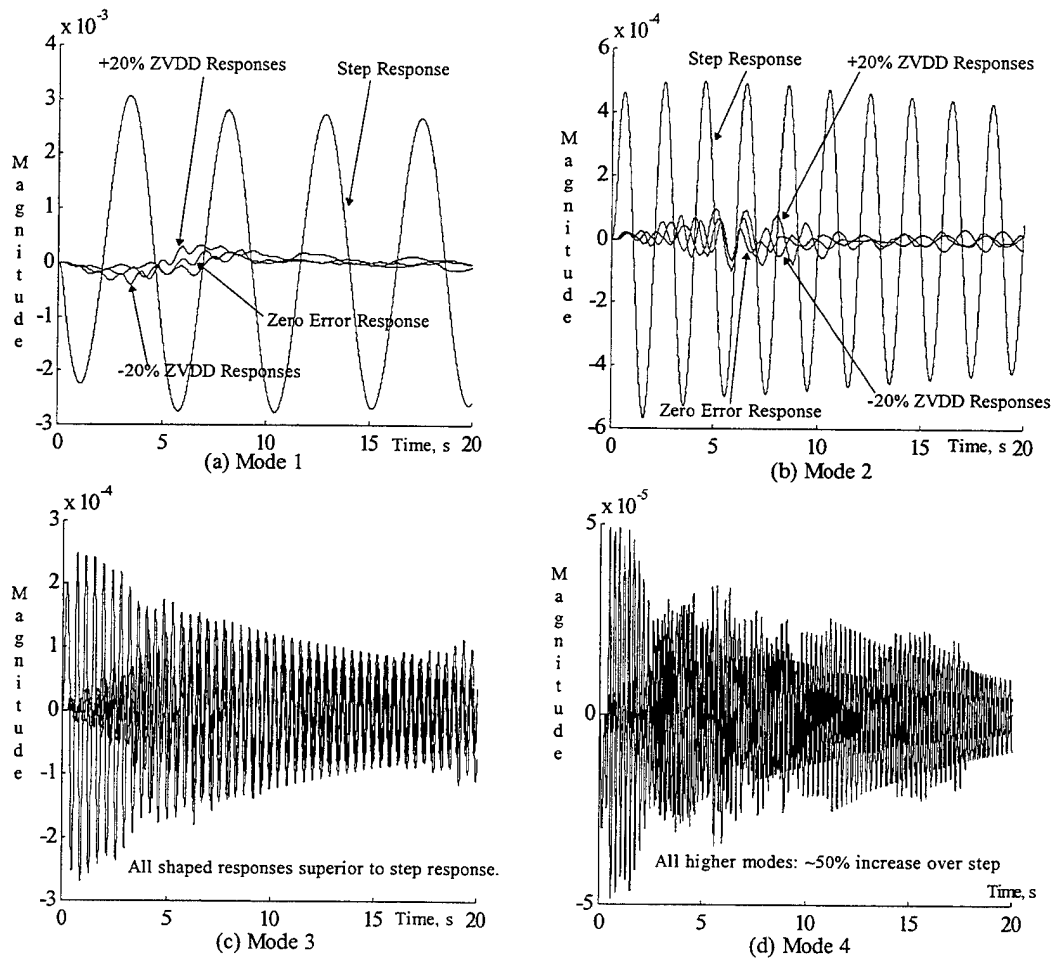
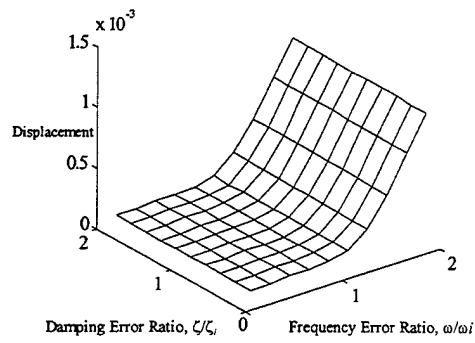
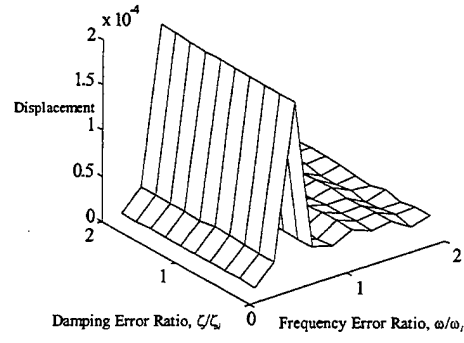


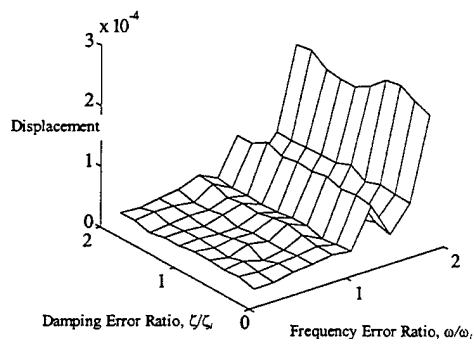
Figure 6.13
ZVDD Shaper Robustness to Frequency Uncertainty



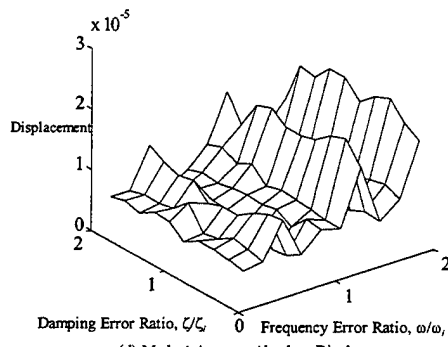
(a) Mode 1 Average Absolute Displacement



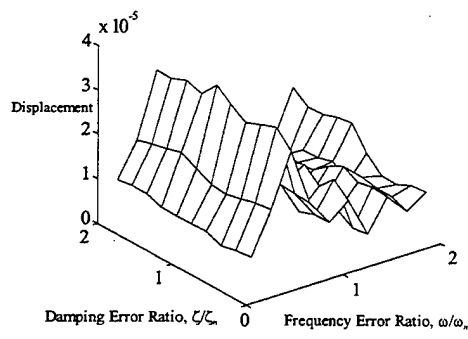
(b) Mode 2 Average Absolute Displacement



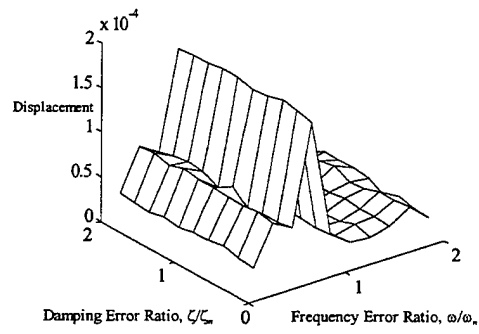
(c) Mode 3 Average Absolute Displacement



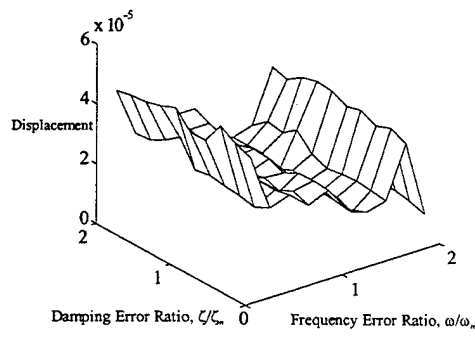
(d) Mode 4 Average Absolute Displacement



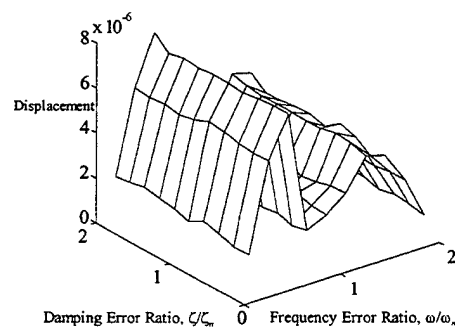
(e) Mode 5 Average Absolute Displacement



(f) Mode 6 Average Absolute Displacement



(g) Mode 7 Average Absolute Displacement



(h) Mode 8 Average Absolute Displacement

Figure 6.14
4-Mode ZVDD Shaper Robustness

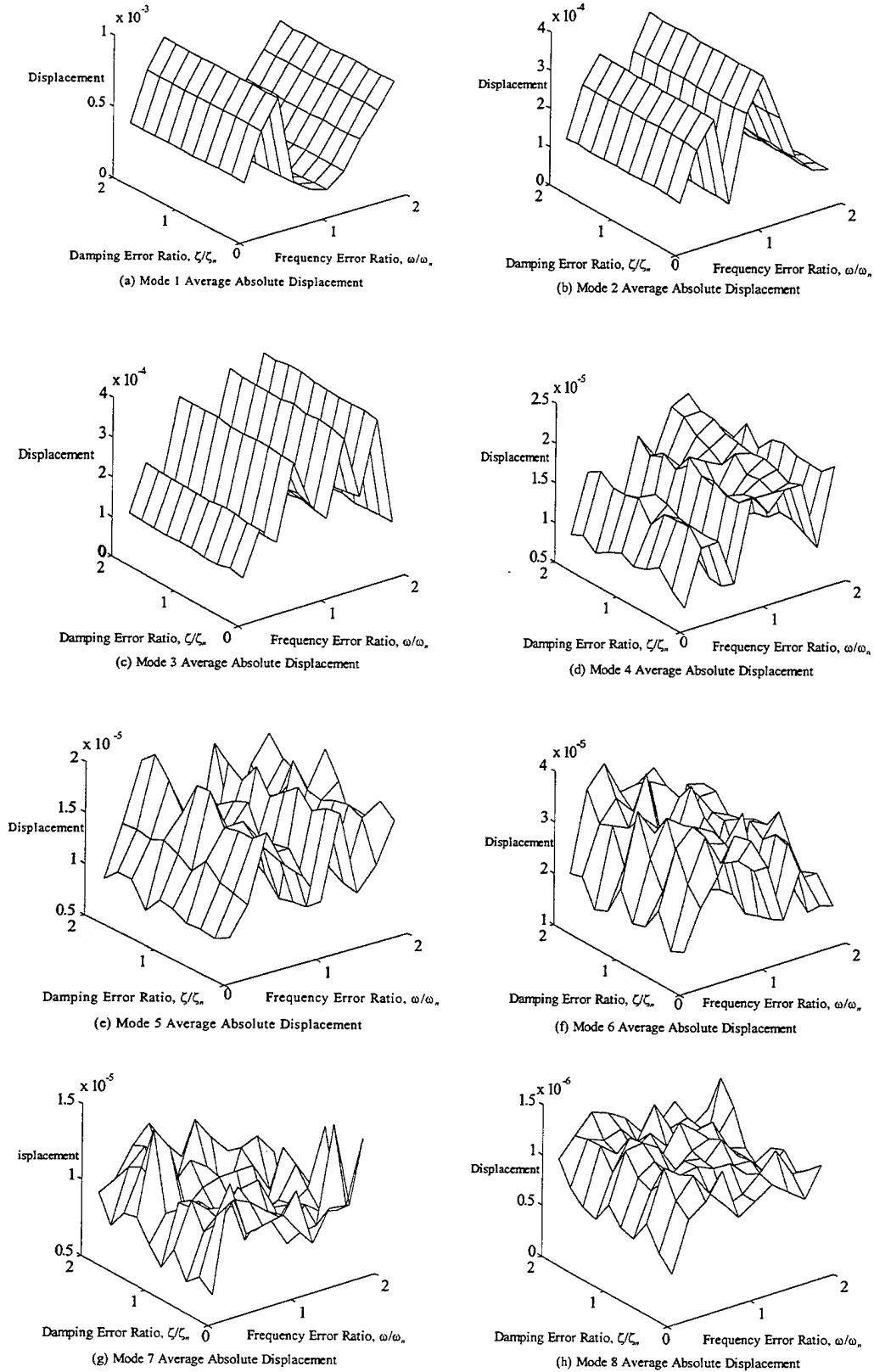


Figure 6.15
1-Mode ZVDD Shaper Robustness

First, the rigid body final stage error is adversely impacted by frequency errors below the actual modal frequency because the shaper command length grows excessively long. A frequency error 50% below the assumed value results in a rapid increase in the maneuver error. Figure 6.12 is illustrative. Shown are the rigid body responses for a 20% over-estimation, nominal plant, and 20% underestimation of the target frequency. As the degree of underestimation increases, the rigid body response becomes increasingly sluggish. The large final stage error indicates that the rigid body has not reached the commanded value in the simulation time.

Second, ZVDD shapers are insensitive to variations in damping whether single- or multiple-mode. Recall that damping ratio is an important determinant of the impulse amplitudes in the shaper pulse train. However, typical flexible spacecraft structures are very lightly damped, so a wide range of damping robustness is not required as long as the system output remains bounded.

Third, the multi-mode shaper is more robust at lower frequencies than the single-mode shaper but entrains more vibration at higher modes. The single-mode shaper retains a significant amount of the step characteristic associated with modal vibrations, but frequency errors in the single-mode shaper do not propagate as strongly to the higher harmonics. If lower modes must be completely eliminated, the multiple-mode shaper does the job at the cost of some entrainment at higher frequencies. With plant uncertainty less than approximately 20%, entrainment is minimal compared to an unshaped command. However, if higher mode entrainment is to be avoided while reducing, but not eliminating, low-mode vibration, a single-mode ZVDD shaper can be quite effective.

Though there is some higher mode entrainment, Figures 6.14 and 6.15 document the relative insensitivity of the entrainment to frequency variations. Absent from the response is the entrainment on the order of 800% reported in CAP shaper research. The reduction in entrainment is a clear improvement over CAP shapers. There are diminishing returns to the vibration reduction possible as mode ratios increase and the spacing in frequency diminishes. Nevertheless, results with a single-mode VA shaper integrated with a PWPF modulator are superior to those of a multi-mode CAP shapers.

E. SUMMARY OF RESULTS

Variable-amplitude actuators have been shown to be superior to bang-bang actuators in terms of minimizing modal excitations (Hailey, 1992; McLelland, 1994). Taking advantage of the PWPF modulator's pseudo-linear operation, the variable amplitude input shaper has been shown to be effective in avoiding residual vibration. Actuating shaped thruster commands with a PWPF modulator yields vibration suppression levels of a linear controller. The rigid body slewing performance remains comparable to that of a bang-bang control. Vibration cancellation is superior to maneuvers with CAP shapers. Entrainment of higher mode vibration is also less pronounced. Finally, the shaped PWPF commands are robust for a wide range of frequency uncertainty. Since current finite element methods can typically identify natural frequencies with less than 10% error, the robustness characteristics of the shaped PWPF commands can allow for changing plant conditions and mass properties.

PWPF modulated control has several distinct advantages. It provides design options of the various controller types. When integrated with a VA input shaper, it can produce slews nearly free of lower mode vibration while using less propellant and fewer thruster cycles than a typical bang-bang controller. Additional benefit is gained from having a tunable range of PWPF parameters, since the modulator settings can be scheduled as a function of plant configuration. Even without modulator tuning, reductions of up to 65% are realized.

VII. CONCLUSIONS AND RECOMMENDATIONS

This thesis presented the first study of PWPF modulated thruster control using the technique of input shaping. An analytical model of the FSS with piezoceramic sensors and actuators was developed. A detailed analysis of the PWPF modulator was performed to determine its suitability to adaptive control. Command input shapers were developed and integrated with the PWPF modulator and comparisons made with regard to shaper type, targeted modes, and maneuver distance. Robustness analyses were performed to show the insensitivity of PWPF modulated input shapers to frequency uncertainty.

A. CONCLUSIONS

The PWPF modulator analyses revealed a narrow but effective tuning range for the modulator parameters. Subsequent investigations using a two-stage gain validated the usefulness of this technique. Use of the recommended design parameter ranges avoids difficulties with excessive phase lag, minimizes thruster cycles and keeps propellant use to a minimum.

Realizing variable amplitude shaped commands with the PWPF modulator is a new technique which capitalizes on the strengths of both bang-bang and linear controllers. The PWPF modulation of variable amplitude shaper commands is especially suited to applications where VA actuators cannot be used. Compared to other methods, this new approach has numerous advantages:

- 1) Simple implementation. The shaper is portable and can be integrated to many flexible systems, whether open- or closed-loop.
- 2) Ease of computation. The shaper commands are dependent only on modal frequency and damping values and are calculated easily from a few vibration constraint equations. Non-linear optimization routines are not required.
- 3) Robustness. The shaped PWPF commands are robust to variations in modal frequency and damping of up to 50 percent. At least four flexible modes can be effectively targeted for cancellation using this method without entraining excessive higher mode vibration.

- 4) Effectiveness. Vibration reductions of up to 96% in the low modes without significant entrainment of higher mode vibration demonstrate the superiority of this method over the CAP-shaping method.
- 5) Economy. Shaped-PWPF commands produce slews nearly free of lower mode vibration while using less propellant than a typical bang-bang controller.

Numerical simulations performed on an eight-mode model of the Flexible Spacecraft Simulator (FSS) in the Spacecraft Research and Design Center (SRDC) at US Naval Postgraduate School (NPS) have demonstrated the efficacy of the variable amplitude shaped PWPF modulator and will serve as a foundation for experimental verification.

B. RECOMMENDATIONS FOR FUTURE RESEARCH

Now that input shaping has been extended by use of the PWPF modulator, several areas for future research remain. In keeping with the design philosophy of the NPS Spacecraft Research and Design Center, these recommendations fall into two general categories of pre-shaping or feed-forward control and active or state-feedback control.

1. Remaining Issues in Input Shaping

While this thesis covers the general theory and shows the simplified design procedure, there are remaining issues in shaper robustness and performance tradeoffs, especially for systems with many tightly-spaced, low-frequency modes. There appears to be a limit on the performance gain realizable with input shaping as the number of targeted modes increases. The shaper type as well as the number and selection of targeted modes play defining roles in this effect. Obtaining detailed understanding of these effects is an area which will require considerable effort.

2. Integrated PWPF Modulated Shaper with Active Vibration Control

Input shaping works well to eliminate the lower mode vibrations, but a different approach should be reserved for the higher modes. For example, an active controller using piezoceramic sensors and actuators with a velocity or positive position feedback system could be used without adding significant complexity or mass to the system.

However, any negative impact of one control type on the other must be identified. For example, a PZT controller has the potential to destabilize the system in certain implementations.

3. Applications for Advanced Control Theory

Optimal control techniques such as Linear Quadratic Gaussian control can also be integrated with the feed-forward methods. Additionally, the shaped-PWPF modulation scheme is an attractive application for fuzzy logic or adaptive control. Specific investigations might include state estimation and real-time system identification to minimize plant uncertainty under dynamic conditions. Design tradeoffs, such as the simplicity of a classical control system against the enhanced performance of the advanced controller, would be key outputs of these research efforts.

4. Experimental Validation

Following a pending upgrade of the Naval Postgraduate School's Flexible Spacecraft Simulator, the results shown in this thesis must be experimentally validated. Specific tasks include implementing a discrete model, imposing hardware limitations such as thruster minimum impulse bit and sampling time restrictions, and analyzing the impact of digital filter time delays on system performance. While this thesis has presented an excellent approach to the problem of slewing flexible spacecraft, there are considerable hurdles to be cleared in real-time implementation. The results of this thesis make shaped-PWPF modulator control a strong candidate for experimental study.

APPENDIX A

Recall the form of the rigid-elastic coupling vector using finite element data:

$$D_i = \sum_{j=1}^N (x_{Fj} \phi_i^{y_j} - y_{Fj} \phi_i^{x_j}) m_j$$

There is one component D_i for each modal frequency. Each D_i is the summation of N terms, where N is the number of nodes (in this case, nine, including the cantilever point).

Using the node and mode shape listings above, the calculations for D_1 are

| Node | Calculation | Subtotal |
|------|--|---------------|
| 1 | $(x_1 \phi_1^{y1} - y_1 \phi_1^{x1}) m_1$ $= [(0.381)(0) - (0)(0)](0.0085)$ | 0 |
| 2 | $(x_2 \phi_1^{y2} - y_2 \phi_1^{x2}) m_2$ $= [(0.533)(-0.0307) - (0)(0)](0.0085 + 0.0085 + 0.455)$ | -7.72e-3 |
| 3 | $(x_3 \phi_1^{y3} - y_3 \phi_1^{x3}) m_3$ $= [(0.686)(-0.0944) - (0)(0)](0.0085 + 0.0085 + 0.455)$ | -3.056e-2 |
| 4 | $(x_4 \phi_1^{y4} - y_4 \phi_1^{x4}) m_4$ $= [(0.838)(-0.1834) - (0)(0)](0.0085 + 0.0085 + 0.455)$ | -7.254e-2 |
| 5 | $(x_5 \phi_1^{y5} - y_5 \phi_1^{x5}) m_5$ $= [(0.991)(-0.3102) - (0)(0)](0.0085 + 0.0085 + 0.91)$ | -2.85e-1 |
| 6 | $(x_6 \phi_1^{y6} - y_6 \phi_1^{x6}) m_6$ $= [(0.991)(-0.3102) - (-0.152)(-0.1561)](0.0085 + 0.0085 + 0.455)$ | -1.563e-1 |
| 7 | $(x_7 \phi_1^{y7} - y_7 \phi_1^{x7}) m_7$ $= [(0.991)(-0.3102) - (-0.305)(-0.333)](0.0085 + 0.0085 + 0.455)$ | -1.93e-1 |
| 8 | $(x_8 \phi_1^{y8} - y_8 \phi_1^{x8}) m_8$ $= [(0.991)(-0.3102) - (-0.457)(-0.5224)](0.0085 + 0.0085 + 0.455)$ | -2.578e-1 |
| 9 | $(x_9 \phi_1^{y9} - y_9 \phi_1^{x9}) m_9$ $= [(0.991)(-0.3102) - (-0.61)(-0.7173)](0.0085 + 0.91)$ | -6.84e-1 |
| | TOTAL: | -1.687 |

APPENDIX B

All computer codes included in this thesis were written by Naval Postgraduate School Space Research and Design Center unless otherwise noted. The code included in this appendix is organized along the lines of the thesis. That is, the initial material applies to the finite element model and the final material corresponds to the shaped PWPF modulator slews and robustness analyses.

A. FINITE ELEMENT MODEL

% FSSmod.m: MAIN MODEL PROGRAM

% This program determines the system response of the Naval Postgraduate School
% Flexible Spacecraft Simulator (FSS). This development includes the rigid body hub,
% the flexible beam, and disturbance torques from the thrusters, piezoceramics, and
% momentum wheel (option). The form of the system matrix can be found in Hailey's
% Naval Postgraduate School 1992 Master's Thesis on the FSS. This more complete
% model includes all the disturbance torques and the piezo equations as forcing
% functions. The called subroutines include "fem.m", which obtains the cantilever
% frequencies and mode shapes for the flexible beam of the FSS. Within "fem.m", the
user is prompted to input the beam parameters and piezoceramic sensor/actuator
locations. "fem.m" calls "femparam.m" to get the beam info.

```
global Amod Bmod Cmod Dmod Amodv % system matrices
global tau Um d h Km Kp hys Kt % PWPF parameters
```

```
% Obtain cantilever modal response, if desired
run=input('Do you need to reenter flexible beam cantilever response (y/n)?','s');
if strcmp(run,'y')==1
    fem
end
```

% DEFINE PARAMETERS

```
Izz=10.49; % Izz=Izzw+Izzf+Izzr (same value as Hailey, p. 98)
zi=.004;
% modal damping factor
omega=omega(1:mods);
% only use frequencies of interest
Tt=[1 zeros(1,mods)];
% Thruster input to center hub
```

```
% Flex Beam controller parameters
% Truncate the piezo sensor and actuator vectors to the number of beam elements
desired.
```



```

Bs=Bs(1:mods);           % sensor vector
Ba=Ba(1:mods);           % actuator vector

% Form the inputs for the System Matrix "A" given on page 12 of Hailey or p.20 of
Watkins
%D=zeros(1,mods); %test the rigid body problem, comment out as required
%omega=zeros(mods,1); % rigid body test (cont'd)
DD=D*D';
Izzo=Izz-DD;
Fi=D.*omega';
Gi=Izzo*omega'+D.*Fi;
Hi=2*zi*omega'.*D;
Ji=2*zi*omega'*Izzo+D.*Hi;

% Form the System Matrix "Asys"

Asys=zeros(2*(mods+1));

% upper right partition
Asys(1:mods+1,mods+2:2*(mods+1))=Izzo*eye(mods+1);

% lower left partition (column 1 is zero).
All=zeros(mods+1);
All(1,1:mods+1)=[0 Fi];
All(2:mods+1,2:mods+1)=-[D'*Fi];
for c=2:mods+1
    All(c,c)=-Gi(c-1);
end
Asys(mods+2:2*(mods+1),1:mods+1)=All;           %install partition in Asys
% lower right partition
Alr=zeros(mods+1);
Alr(1,1:mods+1)=[0 Hi];
Alr(2:mods+1,2:mods+1)=-[D'*Hi];
for c=2:mods+1
    Alr(c,c)=-Ji(c-1);
end
Asys(mods+2:2*(mods+1),mods+2:2*(mods+1))=Alr;   %install partition in Asys

Asys=1/Izzo*Asys;           % premultiply by inertia factor
%fprintf('The system matrix is given by Asys=')
%Asys
% Form the System Input Matrix "Bsys"
Bsys=1/Izzo*[zeros(1,mods+1) 1 -D]';
%fprintf('The system input matrix is given by Bsys=')
%Bsys

```

```

% Form the Observation Matrix "Csys"
%fprintf('The Observation Matrix is give by Csys=')
%Csys=[eye(mods)]

% Form the Transmission Matrix "Dsys"
%fprintf('The Transmission Matrix "Dsys" is identically zero')

% Using the control system variation, given on page 13 of Hailey
Mmod=[Izz D;D' eye(mods)];
Zmod=[0 zeros(1,mods);zeros(mods,1) diag(-2*zi*omegarad(1:mods))];
Kmod=[0 zeros(1,mods);zeros(mods,1) diag(omega)];
[phi1,lambda1]=eig(Kmod,Mmod);
[lambda,phimod,psimod]=eign(Kmod,Mmod); % system freqs and eigvectors
% Find the modal matrix, "modmat" by diagonalizing M and calculating
% modmatrix = eigenmatrix*sqrt((diag M))
MM=phi1'*Mmod*phi1;
for i=1:mods+1
for j=1:mods+1
if abs(MM(i,j))<0.0001
MM(i,j)=0;
end
end
end
modmat=phi1*inv(sqrt(MM));

AA=[zeros(mods+1) eye(mods+1);inv(Mmod)*Kmod zeros(mods+1)];
BB=[zeros(mods+1,1);inv(Mmod)*Tt];
Amod=[zeros(mods+1) eye(mods+1);diag(-lambda) diag(-2*zi*sqrt(lambda))];
Amod1=[zeros(mods+1) eye(mods+1);-lambda1 -2*zi*sqrt(lambda1)];
Bmod=[zeros(mods+1,1); modmat*Tt];
Cmod=eye(2*mods+2); % obs. matrix in modal coordinates
Dmod=zeros(2*mods+2,1);
% Similarity transform to go from modal observation matrix to physical
% coordinates is given by C'=modmat*[state vector]
% Now add damping due to velocity feedback controller
ans=input('Is Velocity feedback piezo control operating? (y/n)','s');
if strcmp(ans,'y')==1
vgain=input('Input the velocity feedback gain');
coeff=zeros(mods+1);
coeff(2:mods+1,2:mods+1)=-2*vgain/ys*Ba*Bs;
coeff=inv(modmat)*coeff*modmat;
Zpiezo=[zeros(mods+1) zeros(mods+1);zeros(mods+1) coeff];
fprintf('The system matrix with velocity feedback is given by')
Amodv=Amod+Zpiezo;
end

```

```

clear ans
ans=input('Is the PWPF modulator installed? (y/n)','s');
if strcmp(ans,'y')==1
    Kp=input('Enter the input gain, Kp= ');
    Km=input('Enter the modulator gain, Km= ');
    Kt=input('Enter the Thruster size, Kt= ');
    tau=input('Enter the modulator time constant, tau= ');
    d=input('Enter the on-threshold, d= ');
    h=input('Enter the off-threshold, h= ');
    Um=input('Enter the trigger output value, Um= ');
    hys=d-h;
end
% Clean up the workspace
clear All Alr E beammass b1a b1s b2a b2s b3a b3s b4a b4s rhop i it c xloc
clear dMa dMs density dof epsilon height length la ls tpa tps Ka Ma j h
clear width ya yloc za zs volume Pa Ps Bssize Basize Ki Mi Ks Ms pmass yloc
clear Ep ans coeff run
% some others that can be cleared at the end...
% clear ys point links m K M Ba Bs
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% FEM.M:
% Flexible Beam Finite Element Model with Controllers %
% LCDR Nick Buck
% Summer 1995
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This program is called by the mfile "FSSmod.m" and uses a finite element model to
% determine the cantilever natural frequencies and mode shapes of a flexible, eight
% element L-shaped beam equipped with piezoelectric sensors on element 2 and
% actuators on beam element 1. The beam is jointed between elements four and five.
% The piezos are mounted on the top and bottom surfaces of the beam.
% Assumptions include: uniform beam density and modulus of elasticity. Moments of
% inertia for point masses are neglected. Boundary conditions are fixed/free.
%
% Three controllers are available for analysis: velocity feedback, positive position
% feedback, and a combination of the two.
%
% After calculating the natural frequencies and mode shapes, this file sorts the output
% and can output the x and y displacements of each node from the undisturbed position.
% Using the displacement and nodal location data, the routine calculates the rigid-
% elastic coupling vector, D, for use in the FSS model. The rigid-elastic coupling vector
% represents the location of any point on the flexible beam, which when crossed with
% central body rotation rate, yields the kinetic energy "cross-term" due to rotation of
% the body-fixed coordinate frame in inertial space.

```

```

%%%%%% Performance Parameters %%%%%%%%%%%%%%%

% obtain beam and piezo parameters if desired
run=input('Do you need to reenter beam/piezo information (y/n)?','s');
if strcmp(run,'y')==1
    femparam
end

%%%%%% Undamped (no controller) Cantilever Response %%%%%%%%%%%%%%%
% Solve the eigenvalue problem using unity modal masses
[omega,psi,phi]=eign(K,M);
omegarad=sqrt(omega);
omegahertz=omegarad/2/pi;
wa=min(omegarad);          % undamped fundamental frequency
xmodes=phi(1:2:15,:);
thetamodes=phi(2:2:16,:);
w=linspace(1,300,300);      %set for first 500 secs of response

% Set up output matrices for viewing modes one at a time

Co1=[1 zeros(1,31)];
Do=[0];
Co2=[0 1 zeros(1,30)];
Co3=[0 0 1 zeros(1,29)];
Co4=[0 0 0 1 zeros(1,28)];
Co5=[0 0 0 0 1 zeros(1,27)];

% Output the frequencies and mode shapes, if desired
ans=input('Display unaugmented natural freqs and modes (y/n)?','s');
if strcmp(ans,'y')==1
    K
    M
    fprintf('The plant (undamped) response is: %f')
    omegahertz=omegarad/2/pi
    xmodes=phi(1:2:15,:)
end

% Sort the modal vectors into the y and x displacements, respectively.
% Maximum number of modes is limited to 16 (8-element model)
mods=input('How many modal displacements would you like to output (1-16)?');

% Form the matrix of y and x displacements, respectively. Every two columns
correspond to a mode.
xylocs=zeros(9,mods*2);

```

```

dis=reshape(xmodes,4,links*4);    % put the xmodes matrix in columns of y and x
displacements.
for it=1:2:mods*2
ymax=dis(4,it); Ymax=[ymax ymax ymax ymax]';
xylocs(2:5,it)=dis(1:4,it);xylocs(6:9,it)=Ymax;
xylocs(6:9,it+1)=dis(1:4,it+1);
end
% append the nodal locations x and y respectively, to the front of the displacement
matrix
xloc=[.381 .533 .686 .838 .991 .991 .991 .991 .991]';
yloc=[0 0 0 0 0 -.152 -.305 -.457 -.610]';
xylocs=[xloc yloc xylocs];

```

```

% Output displacement info if desired
ans=input('Would you like to list the x & y displacements?','s');
if strcmp(ans,'y')==1
    for it=1:mods
fprintf('Mode: %f',it)
fprintf('      x','      y')
xylocs(1:9,2*it+1:2*it+2)
    end
end
ans=input('Would you like to list the x & y node locations?','s');
if strcmp(ans,'y')==1
    fprintf('Node Locations')
    fprintf('      x','      y')
    xylocs(1:9,1:2)
end

```

```

% Lump the beam and point masses at the nodes (point masses are already there).
pointj=reshape(point,2,8);
pointj=[0 pointj(1,:)];          % point masses
mj=zeros(1,9);                  % _ of each element mass goes to the node on either
side
for cc=1:links
mj(cc)=mj(cc)+m/2;
mj(cc+1)=mj(cc+1)+m/2;
end
mj=mj+pointj;                    % lump beam + point masses

```

% Calculate the rigid-elastic coupling vector, D, for use in the FSS model.

```

D=[zeros(1,mods)];
for it=1:mods                    % components of D correspond to each mode
for c=1:9                        % nine nodes in the model

```

```

% Di's =SUM((Xj*PHIy-Yj*PHIx)*mass)
D(it)=D(it)+(xylocs(c,1)*xylocs(c,2*it+1)-xylocs(c,2)*xylocs(c,2*it+2))*mj(c);
end
end
fprintf('The rigid-elastic coupling vector is given by')
D
end

```

```

% FEMPARAM.M %%%%%%%%%

```

```

%
%%%%%%%%%%%%%
%                                LCDR Nick Buck
%                                Summer 1995
%                                FSS Finite Element Modeling
%%%%%%%%%%%%%
%This subroutine determines the parameters of a beam with piezoelectric actuators
% & sensors. Output is used in the finite element program fem.m to determine modal
% response of the beam. The sensor is placed on the beam and the actuator is colocated
% on top of the sensor.
clear;

```

```

%%%%%%%% Beam Parameters %%%%%%%%%

```

```

length=1.2192;           % overall beam length
links=8;                 % number of finite elements
height=0.001575;        % beam height
width=0.0254;           % beam width
density=2.8e3;           % beam density
E=7.2e10;               % Young's Modulus

```

```

%% Calculated Beam Quantities %%

```

```

I=width*height^3/12;     % Moment of inertia for beam elements
volume=length*width*height;
beammass=density*volume;% total beam mass
m=beammass/links; % mass of each beam element
h=length/links;         % length of each element
dof=2*links+2;          % matrix dimensions = degrees of freedom

```

```

%%%%%%%% Piezo Parameters %%%%%%%%%

```

```

d=1.8e-10;% piezoelectric charge coefficient
Ep=6.3e10;% Young's Modulus for material
epsilon=1.5e-8; % permittivity of material

```

```

tps=.25e-3;          % sensor thickness
tpa=0.5e-3;          % actuator thickness
rhop=7.7e3; % density of sensor & actuator material
zs=height/2; % position of sensor piezo from z-axis
za=zs+tps; % position of actuator piezo fm z-axis
Wp=width; % width of piezos
Ps=Wp*tps*Ep*(zs^2+zs*tps+tps^2/3); % Sensor "Stiffness factor"
Pa=Wp*tpa*Ep*(za^2+za*tpa+tpa^2/3); % Actuator "Stiffness factor"

%%%%%% Calculated Piezo Quantities %%%%%%%%%%

%%%%%%%% Sensor %%%%%%%%%

dMs=Wp*tps*rhop; % sensor mass per unit length, h
ys=Wp*h/tps*(epsilon-d^2*Ep); % electric potential

% electro-mechanical coupling coefficients & general coordinates
b1s=0; % q1
b2s=-d*Ep*Wp*(zs+tps/2); % q2
b3s=0; % q3
b4s=d*Ep*Wp*(zs+tps/2); % q4
bs=[b1s b2s b3s b4s];
Bs=zeros(1,dof); % output vector, es=1/ys*B*s*q
ls=input('specify the piezo sensor location. Element #');
Bs(2*ls-1:2*ls+2)=Bs(2*ls-1:2*ls+2)+bs;
Bs=Bs(3:dof); % adjust for fixed end BC
Bssize=max(size(Bs)); % length of Bs

%%%%%%%% Actuator %%%%%%%%%

dMa=Wp*tpa*rhop; % actuator mass per unit length, h
ya=Wp*h/tpa*(epsilon-d^2*Ep); % electric potential

% electro-mechanical coupling coefficients & general coordinates

b1a=0; % q1
b2a=-d*Ep*Wp*(za+tpa/2); % q2
b3a=0; % q3
b4a=d*Ep*Wp*(za+tpa/2); % q4
ba=[b1a b2a b3a b4a]';
Ba=zeros(dof,1); % forcing function, F=-2*ea*B*a
la=input('specify the piezo actuator location. Element #');
Ba(2*la-1:2*la+2)=Ba(2*la-1:2*la+2)+ba;
Ba=Ba(3:dof); % adjust for fixed end BC
Basize=max(size(Ba)); % length of Ba

```

%%%%%%%% Elemental Matrices %%%%%%%%%%

% Beam Stiffness

```
Ki=(E*I/h^3)*[12 6*h -12 6*h
               6*h 4*h^2 -6*h 2*h^2
               -12 -6*h 12 -6*h
               6*h 2*h^2 -6*h 4*h^2];
```

% Beam Mass

```
Mi=(m/420)*[156 22*h 54 -13*h
             22*h 4*h^2 13*h -3*h^2
             54 13*h 156 -22*h
             -13*h -3*h^2 -22*h 4*h^2];
```

% Sensor Stiffness and Mass

```
Ks=Ps/h^3*[12 6*h -12 6*h
            6*h 4*h^2 -6*h 2*h^2
            -12 -6*h 12 -6*h
            6*h 2*h^2 -6*h 4*h^2];
```

```
Ms=dMs*h/420*[156 22*h 54 -13*h
              22*h 4*h^2 13*h -3*h^2
              54 13*h 156 -22*h
              -13*h -3*h^2 -22*h 4*h^2];
```

% Actuator Stiffness and Mass

```
Ka=Pa/h^3*[12 6*h -12 6*h
            6*h 4*h^2 -6*h 2*h^2
            -12 -6*h 12 -6*h
            6*h 2*h^2 -6*h 4*h^2];
```

```
Ma=dMa*h/420*[156 22*h 54 -13*h
              22*h 4*h^2 13*h -3*h^2
              54 13*h 156 -22*h
              -13*h -3*h^2 -22*h 4*h^2];
```

%%%%%%%% Point Mass Input %%%%%%%%%%

```
ans=input('Are there point masses on the beam (y/n)?','s');
```

```
if strcmp(ans,'y')==1
```

```
    for c=1:links
```

```
        fprintf('For node # %f,c)
```

```
        point(2*c-1)=input('What is the point mass?');
```



```

        point(2*c)=0;
    end

    pmass=diag(point); % format to match K,M
    elseif strcmp(ans,'n')==1
        pmass=zeros(2*links);
    end
end

%%%% Global Mass and Stiffness Matrices %%%%%%%%%%%%%%
% Setup inputs from first element

K=zeros(dof,dof);
M=zeros(dof,dof);
K(1:4,1:4)=Ki(1:4,1:4);
M(1:4,1:4)=Mi(1:4,1:4);

% Add additional elements
for i=2:links
    K=K+[zeros(2*(i-1),dof)
        zeros(4,2*(i-1)) Ki zeros(4,dof-2*(i-1)-4)
        zeros(dof-2*(i-1)-4,dof)];
    M=M+[zeros(2*(i-1),dof)
        zeros(4,2*(i-1)) Mi zeros(4,dof-2*(i-1)-4)
        zeros(dof-2*(i-1)-4,dof)];
end

% Remove first two rows/columns to account for fixed end BC
K=K(3:dof,3:dof);
M=M(3:dof,3:dof);

% Add the point masses to the mass matrix
M=M+pmass;

% Add mass of elbow and associated point masses to node 5
% due to constraints W6=W7=W8=W9 (elbow is rigid body)
M(7,7)=M(7,7)+sum(point(links+1:links*2))+4*m;

% Apply constraints to elbow...W5=W6=W7=W8=W9
M(7,9:10)=[0 0];
M(9:10,7)=[0 0]';

% Adjust for elbow U5=W5 (element already accounted for)
K(7,7:10)=K(7,7:10)-Ki(1,1:4);
K(8:10,7)=K(8:10,7)-Ki(2:4,1);

```

```

% Check for piezo actuators
ans=input('Are the piezos installed (y/n)?','s');
if strcmp(ans,'y')==1

    % Add 2 piezo actuators (top & bottom) to element 2
    K(1:4,1:4)=K(1:4,1:4)+2*Ka(1:4,1:4);
    M(1:4,1:4)=M(1:4,1:4)+2*Ma(1:4,1:4);

    % Add 2 piezo sensors (top & bottom) to element 2
    K(1:4,1:4)=K(1:4,1:4)+2*Ks(1:4,1:4);
    M(1:4,1:4)=M(1:4,1:4)+2*Ms(1:4,1:4);

end
end

```

B. SYSTEM MODEL

% **EIGN.M**: SOLVES THE EIGENVALUE PROBLEM AND SORTS THE MODAL VECTORS

```
function [Lambda,Phi,Psi]=eign(A,B)

%   eign: Solve the generalized eigenvalue problem.
%       Order & normalize the eigenvectors.
%
%   [Lambda,Phi,Psi]=eign(A,B)
%
%        $A x = [Lambda] B x$ 
%   with the special ('structural') normalizations:
%        $\Phi(i)' B \Phi(i) = 1$ 
%        $\Psi(i)' B \Psi(j) = \text{kroncker delta}(i,j)$ 
%   where
%        $\Phi(i)$  -- i-th right eigenvector
%        $\Psi(i)$  -- i-th left eigenvector
%       ' -- transpose
%
%   Caution: real mode --  $\text{imag}(Lambda) < 1.e-7$ 
%
%   reference: Junkins & Kim, Dyn. & Ctrl of Structures, ch. 2,4
%
%   programmed by Youdan Kim
%       Dept. of Aerospace Engineering
%       Texas A&M University
%
%   revised date : May 24, 1989
%       APR. 25, 1991
%       Dec. 16, 1991
n=max(size(A)); Lambda=zeros(n,1); Phi=zeros(n); Psi=zeros(n);
%
%
%   Solve Left and Right Eigenvalue Problem
%
[VR,DR]=eig(A,B); [VL,DL]=eig(A',B');
kr=zeros(n,1); kc=kr; er=kr; ec=kr;
%
%
%   Sort Right Eigenvectors and Eigenvalues
%
indr=0; indc=0;
```

```

for i=1:n;
    if abs(imag(DR(i,i))) <= 1.e-7;
        indr=indr+1; kr(indr)=i; er(indr)=DR(i,i);
    elseif imag(DR(i,i)) > 1.e-7;
        indc=indc+1; kc(indc)=i; ec(indc)=DR(i,i);
    end
end
er=real(er(1:indr)); ec=ec(1:indc);
ind=1; [lr,krn]=sort(er); [lc,kcn]=sort(imag(ec));
for i=1:indr+indc;
    if i <= indr;
        Phi(:,i)=real(VR(:,kr(krn(i))));
        Lambda(i)=real(DR(kr(krn(i)),kr(krn(i))));
        ind=ind+1;
    else
        ii=i-indr;
        Phi(:,ind)=VR(:,kc(kcn(ii)));
        Phi(:,ind+1)=conj(Phi(:,ind));
        Lambda(ind)=DR(kc(kcn(ii)),kc(kcn(ii)));
        Lambda(ind+1)=conj(Lambda(ind));
        ind=ind+2;
    end
end
end
%

% Sort Left Eigenvectors
%
indr=0; indc=0;
for i=1:n;
    if abs(imag(DL(i,i))) <= 1.e-7;
        indr=indr+1; kr(indr)=i; er(indr)=DL(i,i);
    elseif imag(DL(i,i)) > 1.e-7;
        indc=indc+1; kc(indc)=i; ec(indc)=DL(i,i);
    end
end
er=real(er(1:indr)); ec=ec(1:indc);
ind=1; [lr,krn]=sort(er); [lc,kcn]=sort(imag(ec));
for i=1:indr+indc;
    if i <= indr;
        Psi(:,i)=real(VL(:,kr(krn(i))));
        ind=ind+1;
    else
        ii=i-indr;
        Psi(:,ind)=VL(:,kc(kcn(ii)));
        Psi(:,ind+1)=conj(Psi(:,ind));
    end
end

```

```

        ind=ind+2;
    end
end
%

%   Normalize Right and Left Eigenvectors
%

for i=1:n;
    xi=Phi(:,i);
    yi=Psi(:,i);
    sc1=conj(xi')*B*xi;    Phi(:,i)=Phi(:,i)/sqrt(sc1);
    sc2=conj(yi')*B*Phi(:,i); Psi(:,i)=Psi(:,i)/sc2;
end

```

C. PWPF MODULATOR CODES

SCHMIDT TRIGGER IMPLEMENTATION FOR FSS MODEL

```
function [sys,x0] = schmtmf(t,x,u,flag,U,Eon,Eoff)
% SCHMTMF Implements a Schmidt Trigger,
% Control Algorithm
% The global variable FLG must be declared
% in the work space and initialized to 0
% A Zero-Order Hold Module must be placed on the
% Output of the Schmidt Trigger and set to the
% simulation time step
global FLG
if abs(flag) == 1
    sys = [];
elseif flag == 3
    if FLG == 0
        if u > Eon
            FLG = 1; sys = 1.0*U;
        elseif u < -Eon
            FLG = -1; sys = -1.0*U;
        else
            FLG = 0; sys = 0.0*U;
        end
    elseif FLG == 1
        if u < Eoff & u >= -Eon
            FLG = 0;
            sys = 0.0*U;
        elseif u < Eoff & u < -Eon
            FLG = -1;
            sys = -1.0*U;
        else
            FLG = 1;
            sys = 1.0*U;
        end
    elseif FLG == -1
        if u > -Eoff & u <= Eon
            FLG = 0;
            sys = 0.0*U;
        elseif u > -Eoff & u > Eon
            FLG = 1;
            sys = 1.0*U;
        else
            FLG = -1;
            sys = -1.0*U;
        end
    end
end
```

```

else
    FLG = 0;
    sys = 0.0*U;
end
elseif flag == 0
    FLG = 0;
    sys = [0,0,1,1,0,1];
else
    sys = [];
end

```

%%
FUNCTION TON: COMPUTES THE THRUSTING TIME FOR THE PWPF MODULATOR

```

function [ont,onnum] =ton(time,xx)
    markton=0;
    thruster=abs(xx);
    onnum=0;
    tton=0;
    ont=0;
    [len,wid]=size(time);
    for k=1:1:(len-1),
        if thruster(k+1)-thruster(k)>0.0;
            tton=time(k+1);
            onnum=onnum+1;
            %fprintf('ton=%f\n\n',tton);
            markton=1;
        end;
        if thruster(k+1)-thruster(k)<0.0;
            toff=time(k+1);
            ont=ont+(toff-tton);
            markton=0;
        end;
    end;
    ont=ont+markton*(time(len)-tton);

    %fprintf('on time=%f\n\n',ont);
    %fprintf('number of firing=%f\n\n',onnum);

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
BATCHK: RUNS THE PWPF SLEWING SIMULATIONS AS A BATCH FILE
%batchk.m

```

```

clear mm nn VKp Vtau mover settl fer onnumb ontime;

for n=1:1:8;
    eval(['clear vibq' int2str(n)]); %clear vibq1,...,vibq8
end

for mm=[1:1:30]
    for nn=[1:1:30]
        Kp=1+mm;
        VKp(mm)=Kp;
        tau=0.01+0.025*nn;
        Vtau(nn)=tau;
        rk45('pmodsim',tf,zeros(20,1),[1e-5,1e-5,1e-3]);

        %calculate the maximum overshoot
        mover(mm,nn)=max(states(:,1))-slew/57.3;

        %Compute the settling time within 5% relative error
        [len,wid]=size(time);
        theta=states(:,1);
        mark1=tf;
        mark2=0;
        for kk=[1:1:len];
            if abs(theta(kk)-slew/57.3)/(slew/57.3)<0.05 & mark2==0;
                mark1=kk;
                mark2=1;
            end
            if mark2==1 & abs(theta(kk)-slew/57.3)/(slew/57.3)>=0.05;
                mark2=0;
                mark1=tf;
            end
        end
        if mark2==0
            settl(mm,nn)=tf;
            fprintf('    Settling time is not reached\n\n')
        end
        if mark2==1
            settl(mm,nn)=time(mark1);
            %fprintf('    Settling time is %f\n\n', time(mark1))
        end
        %compute the total thruster action - on-time
    end
end

```



```

%and number of firings
thruster=abs(pwpfout(:,3));
onnum=0;
ont=0;
for k=1:(len-1),
    if thruster(k+1)-thruster(k)>0.0001;
        tton=time(k+1);
        onnum=onnum+1;
        %fprintf('ton=%f\n\n',tton);
        mark4=1;
    end;
    if thruster(k+1)-thruster(k)<-0.000001;
        ttoff=time(k+1);
        ont=ont+(ttoff-tton);
        mark4=0;
    end;
end;
ont=ont+mark4*(tf-tton);
ontime(mm,nn)=ont; %total on time
onnumb(mm,nn)=onnum; %number of firings
fprintf('on time=%f\n\n',ontime);
%find the final stage error (average of the absolute
%value of the error during the last 10% simulation
%time)
%[len,wid]=size(time);
%theta=states(:,1);
mark3=0;
for kk=[1:1:len];
    if time(kk)==tf*0.9;
        mark3=kk;
    end;
end;
sum=0;

for kk=[kk:1:len];
    sum=sum+(theta(kk)-slew*pi/180);
end;
%fer=sum/(len-kk+1);
fer(mm,nn)=sum/(len-kk+1);
fprintf('    Final Error is %f\n\n', fer(mm,nn))

%calculate the index for each modal vibration
%the index is simply the average of the absalute value of the
%each modal displacement

```

```

    for n=2:1:9
        eval(['vibq' int2str(n-1) '(mm,nn)=mean(abs(states(:, ' int2str(n) ')));']);
    end

    %save the data into 'data###' matrix
    %eval(['data' int2str(mm) int2str(nn) '=[time states pdout pwpfout command];']);
end
fprintf('    mm is %f\n\n', mm)
end
%for mm=[1:1:3]
    %eval(['dat=data1' int2str(mm) '']);
    %plot(dat(:,1),dat(:,2))
    hold on
%end
figure
mesh(Vtau,VKp,mover)
title('Max Overshoot')
figure
mesh(Vtau,VKp,sett1)
title('Settling Time')

figure
mesh(Vtau,VKp,fer)
title('Final Stage Error')

figure
mesh(Vtau,VKp,ontime)
title('On Time')

figure
mesh(Vtau,VKp,onnumb)
title('Number of Thrusterings')

figure
mesh(Vtau,VKp, vibq1);
title('Average Absolute Displacement of Modal 1');

figure
mesh(Vtau,VKp, vibq2);
title('Average Absolute Displacement of Modal 2');

figure
mesh(Vtau,VKp, vibq3);
title('Average Absolute Displacement of Modal 3');

```

```
figure
mesh(Vtau,VKp, vibq4);
title('Average Absolute Displacement of Modal 4');
```

```
figure
mesh(Vtau,VKp, vibq5);
title('Average Absolute Displacement of Modal 5');
```

```
figure
mesh(Vtau,VKp, vibq6);
title('Average Absolute Displacement of Modal 6');
```

```
figure
mesh(Vtau,VKp, vibq7);
title('Average Absolute Displacement of Modal 7');
```

```
figure
mesh(Vtau,VKp, vibq8);
title('Average Absolute Displacement of Modal 8');
```

```
hold
save 50by50;
end
```

BATCHP.M: RUNS THE MODULATOR PARAMETER VARIATIONS IN BATCH FORMAT

```
%d=0.45; %on threshold
clear d h;
if exist('countkm')==1;
for n=1:1:countkm;
    eval(['clear b' int2str(n)]);
    eval(['clear B' int2str(n)]);
    eval(['clear Tont' int2str(n)]);
    eval(['clear Tofft' int2str(n)]);
    eval(['clear freq' int2str(n)]);
    eval(['clear MF' int2str(n)]);
    eval(['clear dead' int2str(n)]);
    eval(['clear pwmin' int2str(n)]);
    eval(['clear rmax' int2str(n)]);
    eval(['clear Tonc' int2str(n)]);
    eval(['clear onnumb' int2str(n)]);
    eval(['clear MFc' int2str(n)]);
```

```

    eval(['clear Toffc' int2str(n)]);
    eval(['clear Error' int2str(n)]);
end
end

tf=1;
FLG=0;
tau=0.13;

cinput=0.5;
Um=1;
r=cinput;

countd=10; %how many different d you want to run
dmax=1; %max on d
counth=10;
countkm=10;
Kmmax=20;

for j=1:1:countkm;
    Km(j)=Kmmax*(j/countkm);
    for n=1:1:(countd);
        d(n)=dmax*((n)/countd);
        for i=1:1:counth;
            fprintf('j is %f,j);
            fprintf('  n is %f,n);
            h(i)=d(n)*((i)/counth);
            eval(['b' int2str(j) '(n,i)=h(i)/(Km(j)*Um-h(i));']);
            eval(['B' int2str(j) '(n,i)=(Km(j)*r-d(n))/(Km(j)*Um-h(i));']);
            eval(['Tont' int2str(j) '(n,i)=-tau*log(1+h(i)/(Km(j)*(r-Um)-d(n)));']);
            eval(['Tofft' int2str(j) '(n,i)=-tau*log(1-h(i)/(Km(j)*r-d(n)+h(i)));']);
            Tontemp=-tau*log(1+h(i)/(Km(j)*(r-Um)-d(n)));
            Toftime=-tau*log(1-h(i)/(Km(j)*r-d(n)+h(i)));
            eval(['freq' int2str(j) '(n,i)=1/(Tontemp+Toftime);']);
            eval(['MF' int2str(j) '(n,i)=Tontemp/(Toftime+Tontemp);']);
            eval(['dead' int2str(j) '(n,i)=d(n)/Km(j);']);
            eval(['rmax' int2str(j) '(n,i)=Um+(d(n)-h(i))/(Km(j));']);
            eval(['pwwmin' int2str(j) '(n,i)=tau*log(1+h(i)/(Km(j)*Um-h(i)));']);

            rk45('brigpwpf', tf, 0, [1e-5,1e-5,1e-4,0,0,2]);

            [ontime, onnum]=ton(time,pwpfo(:,3));

            fprintf('  On Time is %fn',ontime);
            eval(['Tonc' int2str(j) '(n,i)=ontime;']);

```

```

    eval(['onnumb' int2str(j) '(n,i)=onnum;']);
    eval(['Toffc' int2str(j) '(n,i)=tf-ontime;']);
    eval(['MFc' int2str(j) '(n,i)=ontime/tf;']);
    eval(['Error' int2str(j) '(n,i)=mean(abs(pwpfo(:,1)));']);
end;
end;
end;

```

```

figure;
orient tall;
for j=1:1:countkm;
    subplot(5,2,j);
    eval(['mesh(h, d, b' int2str(j) ',MF' int2str(j) ')']);
    colormap(datamap);
    if j<=2;
        title('b ');
    end; axis([0 1 0 1 0 1])
    text(0.8,0.8,['Km=', int2str(Km(j))],'units','normalized');
    if j>=9;
        xlabel('Eoff');
        ylabel('Eon');
    end
end
end

```

```

figure;
orient tall;
for j=1:1:countkm;
    subplot(5,2,j);
    eval(['mesh(h, d, B' int2str(j) ')']);
    colormap(datamap);
    if j<=2;
        title('B ');
    end; axis([0 1 0 1 0 1])
    text(0.8,0.8,['Km=', int2str(Km(j))],'units','normalized');
    if j>=9;
        xlabel('Eoff');
        ylabel('Eon');
    end
end
end

```

```

figure;
orient tall;
for j=1:1:countkm;
    subplot(5,2,j);
    eval(['mesh(h, d, MF' int2str(j) ')']);

```

```

colormap(datamap);
if j<=2;
    title('MF ');
end; axis([0 1 0 1 0 1])
text(0.8,0.8,['Km=', int2str(Km(j))],'units','normalized');
if j>=9;
    xlabel('Eoff');
    ylabel('Eon');
end
end

```

```

figure;
orient tall;
for j=1:1:countkm;
    subplot(5,2,j);
    eval(['mesh(h, d, freq' int2str(j) ')']);
    colormap(datamap);
if j<=2;
    title('Frequency ');
end;
axis([0 1 0 1 0 2500])
text(0.8,0.8,['Km=', int2str(Km(j))],'units','normalized');
if j>=9;
    xlabel('Eoff');
    ylabel('Eon');
end
end

```

```

figure;
orient tall;
for j=1:1:countkm;
    subplot(5,2,j);
    eval(['mesh(h, d, Tonc' int2str(j) ',onnumb' int2str(j) ')']);
    %colormap(datamap);
if j<=2;
    title('Ontime-Simulation');
end;
axis([0 1 0 1 0 1])
text(0.8,0.8,['Km=', int2str(Km(j))],'units','normalized');
if j>=9;
    xlabel('Eoff');
    ylabel('Eon');
end
end

```

```

figure;
orient tall;
for j=1:1:countkm;
subplot(5,2,j);
eval(['mesh(h, d, onnumb' int2str(j) ',Tonc' int2str(j) ')']);
%colormap(datamap);
if j<=2;
title('OnNumber-Simulation');
end;
%axis([0 1 0 1 0 4000])
text(0.8,0.8,['Km=', int2str(Km(j))],'units','normalized');
if j>=9;
xlabel('Eoff');
ylabel('Eon');
end
end

```

```
clear Tontemp Tofftemp;
```

```
%%%%%%%%%
```

PLOTP: PLOTS THE OUTPUT FROM BATCHP IN VARIOUS SUBPLOTS

```

datamap=hsv;
datamap=datamap(11:64,:);

```

```

figure;
orient tall;
for j=1:1:countkm;
subplot(5,2,j);
eval(['mesh(h, d, b' int2str(j) ',MF' int2str(j) ')']);
colormap(datamap);
if j<=2;
title('b ');
end; axis([0 1 0 1 0 1])
text(0.8,0.8,['Km=', int2str(Km(j))],'units','normalized');
if j>=9;
xlabel('Eoff');
ylabel('Eon');
end
end

```

```

figure;
orient tall;
for j=1:1:countkm;
subplot(5,2,j);
eval(['mesh(h, d, B' int2str(j) ')']);

```

```

colormap(datamap);
if j<=2;
    title('B ');
end; axis([0 1 0 1 0 1])
text(0.8,0.8,['Km=', int2str(Km(j))],'units','normalized');
if j>=9;
    xlabel('Eoff');
    ylabel('Eon');
end
end
end

```

```

figure;
orient tall;
for j=1:1:countkm;
    subplot(5,2,j);
    eval(['mesh(h, d, MF' int2str(j) ')']);
    colormap(datamap);

```

```

if j<=2;
    title('MF ');
end; axis([0 1 0 1 0 1])
text(0.8,0.8,['Km=', int2str(Km(j))],'units','normalized');

```

```

if j>=9;
    xlabel('Eoff');
    ylabel('Eon');
end
end
end

```

```

figure;
orient tall;
for j=1:1:countkm;
    subplot(5,2,j);
    eval(['mesh(h, d, freq' int2str(j) ')']);
    colormap(datamap);

```

```

if j<=2;
    title('Frequency ');
end;
axis([0 1 0 1 0 2500])
text(0.8,0.8,['Km=', int2str(Km(j))],'units','normalized');
if j>=9;
    xlabel('Eoff');
    ylabel('Eon');

```



```

    end
end

figure;
orient tall;

for j=1:1:countkm;
    subplot(5,2,j);
    eval(['mesh(h, d, Tonc' int2str(j) ',onnumb' int2str(j) ')']);
    %colormap(datamap);

    if j<=2;
        title('OnTime-Simulation');
    end;
    axis([0 1 0 1 0 1])
    text(0.8,0.8,['Km=', int2str(Km(j))],'units','normalized');

    if j>=9;
        xlabel('Eoff');
        ylabel('Eon');
    end
end

figure;
orient tall;

for j=1:1:countkm;
    subplot(5,2,j);
    eval(['mesh(h, d, onnumb' int2str(j) ',Tonc' int2str(j) ')']);
    %colormap(datamap);

    if j<=2;
        title('OnNumber-Simulation');
    end;

    %axis([0 1 0 1 0 4000])
    text(0.8,0.8,['Km=', int2str(Km(j))],'units','normalized');

    if j>=9;
        xlabel('Eoff');
        ylabel('Eon');
    end
end

```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

FREQ.M: DYNAMIC ANALYSIS FOR PWPF MODULATOR

```
%find how input frequency and time constant affect the PWPF
```

```
km=4.5;Um=1;d=0.45;h=0.15;
```

```
%tau=0.15;
```

```
if exist('Y')==1;
```

```
clear Y Pyy f Pxx X fx ;
```

```
%clear tton kon toff koff ont;
```

```
end;
```

```
countfin=4;
```

```
counttau=1;
```

```
taumax=0.15;
```

```
finmax=20*pi;
```

```
a=5120;
```

```
b=2560;
```

```
for i=1:1:countfin
```

```
    fin=finmax*(i/countfin);
```

```
    Vfin(i)=fin;
```

```
    for j=1:1:counttau;
```

```
        tau=taumax*(j/counttau);
```

```
        Vtau(j)=tau;
```

```
rk45('brigpwpf', (2*pi/fin+0.01),0, [1e-5,1e-5,1e-4,0,0,2]);
```

```
    Y=fft(pwpfo(:,3),a);
```

```
    Pyy=Y.*conj(Y)/a;
```

```
    f=10000*(0:(b-1))/a;
```

```
    %figure
```

```
    subplot(2,2,i)
```

```
    plot(f,Py(1:(b)), 'green')
```

```
    title(['Time constant=' num2str(tau) '   Input Frequency=' num2str(fin/(2*pi))])
```

```
    axis([0 100 0 200])
```

```
    hold on
```

```
    X=fft(datain,a);
```

```
    Pxx=X.*conj(X)/a;
```

```
    fx=10000*(0:(b-1))/a;
```

```
    plot(fx,Pxx(1:(b)), 'red')
```

```
    hold
```

```
end;
```

```
end;
```

%%%

LINEAR: ASSESS LINEARITY OF THE PWPF MODULATOR

%study linear opeartion range of a PWPF

```

if exist('Vinput')==1;
    clear Vinput Vd Tonc onnumb MFc averror pwpfo time;
end
tf=1;
Um=1;
Km=4.5;
tau=0.13;
countin=10;
inputmax=1.2;
countd=10;
dmax=0.9;
for i=1:1:countin;
    input=inputmax*(i/countin);
    Vinput(i)=input;
    for j=1:1:countd;
        d=dmax*(j/countd);
        Vd(j)=d;
        h=0.5*d; %0.5*d;
        fprintf('i is %f,i);
        fprintf(' j is %f,j);
        rk45('brigpwpf, tf,0, [1e-5,1e-5,1e-4, 0, 0,2]);
        [Tonc(i,j),onnumb(i,j)]=ton(time,pwpfo(:,3));
        MFc(i,j)=Tonc(i,j)/tf;
        fprintf(' MFc is %fn', MFc(i,j));
        averror(i,j)=mean(abs(pwpfo(:,1)));
    end;
end;
for j=1:1:countd;
    plot(Vinput,MFc(:,j));
    title('MF');
    axis([0 1 0 1]);
    hold on
end;
hold
figure;
for j=1:1:countd;
    plot(Vinput,onnumb(:,j));
    title('Thruster Firing Number');
    axis([0 1.2 0 150]);
    hold on
end;

```

```

hold
figure;
for j=1:1:countd;
    plot(Vinput,averror(:,j));
    axis([0 1.2 0 1]);
    title('error');
    hold on
end;
hold

```

```

for j=1:1:countd;
    plot(Vinput,MFc(:,j));
    title('MF');
    axis([0 1 0 1]);
    hold on
end;
hold

```

```

figure;

for j=1:1:countd;
    plot(Vinput,onnumb(:,j));
    title('Thruster Firing Number');
    axis([0 1.2 0 150]);
    hold on
end;
hold

```

```

figure;

for j=1:1:countd;
    plot(Vinput,averror(:,j));
    axis([0 1.2 0 1]);
    title('error');
    hold on
end;
hold

```

%%%%%%%%%

LINK.M: ASSESS THE IMPACT OF KM ON MODULATOR LINEARITY

%study linear operation range of a PWPF with the change of Km

```
if exist('Vinput')==1;
    clear Vinput Vd Tonc onnumb MFc averror;
end

tf=1;

d=0.45;
h=0.15;

Um=1;

tau=0.13;

countin=12;
inputmax=1.2;

countkm=10;
kmmax=20;
for i=1:1:countin;
    input=inputmax*(i/countin);
    Vinput(i)=input;
    for j=1:1:countkm;
        km=kmmax*(j/countkm);
        Vkm(j)=km;
        fprintf('i is %f,i);
        fprintf(' j is %f,j);
        rk45('brigpwpf', tf,0, [1e-5,1e-5,1e-4,0,0,2]);
        [Tonc(i,j),onnumb(i,j)]=ton(time,pwpfo(:,3));
        MFc(i,j)=Tonc(i,j)/tf;
        fprintf(' MFc is %f\n', MFc(i,j));
        averror(i,j)=mean(abs(pwpfo(:,1)));
    end;
end;
for j=1:1:countkm;
    if j==1;
        plot(Vinput,MFc(:,j),'red');
        axis([0 1.2 0 1]);
        title('MF');
        hold on;
    elseif j==countkm;
```

```

    plot(Vinput,MFc(:,j),'green');
else
    plot(Vinput,MFc(:,j));
end;
end;
hold
figure;
for j=1:1:countkm;
    if j==1;
        plot(Vinput,onnumb(:,j),'red');
        axis([0 1.2 0 150]);
        title('Number of Thruster Firing');
        hold on;
    elseif j==countkm;
        plot(Vinput,onnumb(:,j),'green');
    else
        plot(Vinput,onnumb(:,j));
    end;
end;
hold
figure;
for j=1:1:countkm;
    plot(Vinput,averror(:,j));
    axis([0 1.2 0 1]);
    title('Error');
    hold on
end;
hold

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% PLOTK: PLOT OUTPUT FROM LINK.M

```

```

for j=1:1:countkm;

    if j==1;
        plot(Vinput,MFc(:,j),'red');
        axis([0 1.2 0 1]);
        title('MF');
        hold on;

    elseif j==countkm;
        plot(Vinput,MFc(:,j),'green');
    else
        plot(Vinput,MFc(:,j));
    end;

```

```

end;
%hold

figure;

for j=1:1:countkm;

    if j==1;
        plot(Vinput,onnumb(:,j),'red');
        axis([0 1.2 0 150]);
        title('Number of Thruster Firing');
        hold on;

    elseif j==countkm;
        plot(Vinput,onnumb(:,j),'green');

    else
        plot(Vinput,onnumb(:,j));
    end;
end;

%hold

figure;

for j=1:1:countkm;
    plot(Vinput,averror(:,j));
    axis([0 1.2 0 1]);
    title('Error');
    hold on
end;
hold

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
LINTAU.M
%study linear opeartion range of a PWPF with the change of tau

if exist('Vinput')==1;
    clear Vinput Vd Tonc onnumb MFc averror;
end

tf=1;

km=4.5;

```

```

d=0.45;
h=0.15;

Um=1;

%tau=0.13;

countin=12;
inputmax=1.2;

counttau=10;
taumax=0.5;

for i=1:1:countin;
    input=inputmax*(i/countin);
    Vinput(i)=input;
    for j=1:1:counttau;
        tau=taumax*(j/counttau);
        Vtau(j)=tau;
        fprintf('i is %f,i);
        fprintf(' j is %f,j);
        rk45('brigpwpf', tf,0, [1e-5,1e-5,1e-4,0,0,2]);
        [Tonc(i,j),onnumb(i,j)]=ton(time,pwpfo(:,3));
        MFc(i,j)=Tonc(i,j)/tf;
        fprintf(' MFc is %f\n', MFc(i,j));
        aerror(i,j)=mean(abs(pwpfo(:,1)));
    end;
end;

figure;
for j=1:1:counttau;
    if j==1;
        plot(Vinput,MFc(:,j),'red');
        axis([0 1.2 0 1]);
        title('MF');
        hold on;
    elseif j==counttau;
        plot(Vinput,MFc(:,j),'green');
    else
        plot(Vinput,MFc(:,j),'blue');
    end;
end;
hold

figure;

```



```

for j=1:1:counttau;
    if j==1;
        plot(Vinput,onnumb(:,j),'red');
        axis([0 1.2 0 150]);
        title('Number of Thruster Firing');
        hold on;
    elseif j==counttau;
        plot(Vinput,onnumb(:,j),'green');
    else
        plot(Vinput,onnumb(:,j),'blue');
    end;
end;

```

hold

```

figure;
for j=1:1:counttau;
    if j==1;
        plot(Vinput,averror(:,j),'red');
        axis([0 1.2 0 1]);
        title('Error');
        hold on;
    elseif j==counttau;
        plot(Vinput,averror(:,j),'green');
    else
        plot(Vinput,averror(:,j),'blue');
    end;
end;
hold

```

%%
PLOTTAU: PLOT OUTPUT FROM LINTAU

```

figure;
for j=1:1:counttau;
    if j==1;
        plot(Vinput,MFc(:,j),'red');
        axis([0 1.2 0 1]);
        title('MF');
        hold on;
    elseif j==counttau;
        plot(Vinput,MFc(:,j),'green');
    else
        plot(Vinput,MFc(:,j),'blue');
    end;
end;

```

```

    end;
end;
hold

figure;
for j=1:1:counttau;
    if j==1;
        plot(Vinput,onnumb(:,j),'red');
        axis([0 1.2 0 150]);
        title('Number of Thruster Firing');
        hold on;
    elseif j==counttau;
        plot(Vinput,onnumb(:,j),'green');
    else
        plot(Vinput,onnumb(:,j),'blue');
    end;
end;
end;

```

```

hold

```

```

figure;
for j=1:1:counttau;
    if j==1;
        plot(Vinput,averror(:,j),'red');
        axis([0 1.2 0 1]);
        title('Error');
        hold on;
    elseif j==counttau;
        plot(Vinput,averror(:,j),'green');
    else
        plot(Vinput,averror(:,j),'blue');
    end;
end;
end;

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

%PHAS.M

```

```

%function [inphase,outphase,tmid,onnum,tont] =phas(time,input,thruster)
function [inphase,outphase,tmid,onnum,tont] =phas(time,input,thruster)

```

```

%clear inphase outphase tmid onnum tont;
%clear tton kon toff koff ont;
%input=datain;
%thruster=pwpfo(:,3);

```

```

[len,wid]=size(time);

```

```

%find the first zero-crossing point for input
%for sine (zero phase shift), going up here
k1=1;
while input(k1)~=0;
    k1=k1+1;
end;

%find the next zero-crossing point for input
%for sine (zero phase shift), going down here

k2=k1+2;
while sign(input(k2))==sign(input(k2+1)) & k2<len;
    k2=k2+1;
end;
%for sine input, assume the peak point is a mark of phase
inphase=(time(k1)+time(k2))/2;

%calculate the ontime,onnumber,the phase
markton=0;
onnum=0;
tton(1)=0;
ont(1)=0;
%tont=0;

for k=k1:1:k2,
    if abs(thruster(k+1))-abs(thruster(k))>0.0;
        onnum=onnum+1;
        tton(onnum)=time(k+1); %time start to fire
        kon(onnum)=k+1;
        %fprintf('ton=%f\n\n',tton);
        markton=1;
    end;
    if abs(thruster(k+1))-abs(thruster(k))<0.0;
        ttoff(onnum)=time(k+1); %time stop firing
        koff(onnum)=k+1;
        ont(onnum)=(ttoff(onnum)-tton(onnum)); %
        markton=0;
    end;
end;

if markton==1 & kon(onnum)<k2,
    for k=(k2+1):1:ceil(1.5*k2);
        if (abs(thruster(k+1))-abs(thruster(k)))<0.0;
            ttoff(onnum)=time(k+1); %time stop firing

```

```

        koff(onnum)=k+1;
        ont(onnum)=(ttoff(onnum)-tton(onnum)); %
        markton=0;
    end;
end;
end;

%compute the total on time
if onnum>=1;
    tont=sum(ont);
end
    tont=tont+markton*(time(len)-tton(onnum)); %total on time

%calculate the output phase --1st method
tmid=(tton(1)+ttoff(onnum))/2;
%calculate the output phase --2nd method
tcum=0;
nn=kon(1);
while tcum<(tont/2) & nn<koff(onnum);
    nn=nn+1;
    tcum=tcum+thruster(nn)*(time(nn)-time(nn-1));
end;
    outphase=time(nn+1);
end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%PLOTREQ.M: PLOTS OUTPUT FROM REQ.M

datamap=hsv;
datamap=datamap(11:64,:);

figure;
mesh(Vfin,Vtau,shift1,tont)
title('phase shift1 (cum), Color--tonot')
xlabel('input frequency');
ylabel('Time constant');
colormap(datamap);

figure;
mesh(Vfin,Vtau,shift2,tont)
title('phase shift2 (mid), Color--tonot')
xlabel('input frequency');
ylabel('Time constant');
colormap(datamap);

```

```
figure;  
mesh(Vfin,Vtau,tont,onnum)  
title('tont, Color--number of firing')  
xlabel('input frequency');  
ylabel('Time constant');  
colormap(datamap);
```

```
figure;  
mesh(Vfin,Vtau,onnum, tont)  
title('number of firing, Color--tonot')  
xlabel('input frequency');  
ylabel('Time constant');  
colormap(datamap);
```

D. INPUT SHAPING CODES

%%
SHAPIN.M: COMPUTES INPUT SHAPER PULSE TRAINS

% shapin.m

% computes the firing times and pulse amplitudes for the input
% shaper using the algorithm detailed in Singer and Seering.
% Zero vibration derivative constraint is used (ZVD).

% Requirements: lambda, mods, and zi must be in the workspace via
% the initialization program "fssmod" or otherwise specified.

% lambda: the vector of squared natural frequencies for the system
% mods: the number of flexible modes from the fssmod routine.
% zi: the damping ratio for each of the flexible modes. If no velocity
% feedback used (piezos), assume constant zeta for all modes and
% zi is then a scalar. This code assumes constant zeta.

if exist('ti')==1;
clear ti imp vibmods modnum;
end

fprintf(['There are ',num2str(mods), ' modes.%\n'])
vibs=input('How many modes do you want to cancel?');
for j=1:vibs;
modnum(j)=input(['No.' num2str(j) ' mode you want to cancel is?']);
end

for j=1:vibs;
vibmods(j)=sqrt(lambda(modnum(j)+1));
end

fprintf('\n The modes you want to cancel are %f\n',vibmods)

deltat=(pi/sqrt(1-zi^2))./vibmods'

%ti=zeros(3,4);
%imp=zeros(3,4);

go=input('What is the command initiation time (seconds)?')
% for each mode, calculate the firing time and pulse amplitude

```

% based on the type of shaper. The delta t's are the same for both

type=input('Choose ZVD or ZVDD ("D" or "DD")','s');
if strcmp(type,'D')~=1 & strcmp(type,'DD')~=1
    type=input('Try that again. What type: D or DD?','s');
end
if strcmp(type,'DD')==1
    % the initial pulse occurs at maneuver time t0=go

    % normalize the pulse amplitudes to unity. If necessary, the shaped
    % command can be amplified such that it makes maximum use of actuator
    % authority or in case of PWPF, is larger than the deadband and remains
    % in the linear range.

    ai=exp(-zi*pi/sqrt(1-zi^2));          % same damping ratio for all modes
    sizer=1+3*ai+3*ai^2+ai^3;
    for j=1:1:vibs
        ti(j,:)=[go go+deltat(j) go+2*deltat(j) go+3*deltat(j)]; % four impulse sequence
        imp(j,:)=[1/sizer 3*ai/sizer 3*ai^2/sizer ai^3/sizer];
    end

else if strcmp(type,'D')==1

    % the initial pulse occurs at maneuver time t0=go
    % normalize the pulse amplitudes to unity. If necessary, the shaped
    % command can be amplified such that it makes maximum use of actuator
    % authority or in case of PWPF, is larger than the deadband and remains
    % in the linear range.

    ai=exp(-zi*pi/sqrt(1-zi^2));          % same damping ratio for all modes
    sizer=1+2*ai+ai^2;
    for j=1:1:vibs
        ti(j,:)=[go go+deltat(j) go+2*deltat(j) 0]; % three impulse sequence
        imp(j,:)=[1/sizer 2*ai/sizer ai^2/sizer 0];
    end

end

end

% set the impulse sizes for simulink
% Ion=[1/sizer ain*ones(1,vibs)];
% Ioff=[-1/sizer -ain*ones(1,vibs)];

% run simulink code to generate the pulse train and command vectors. The simulink
% code returns the variables "impin", "command", and tiempo (time) to the workspace.
% rk45('shapesim',10,[],[1e-5,1e-5,1e-3]);

```

```

% obtain the convolution of the input shaper and the command. The length of the
% overall shaped command is the length of the impulse train+length of the command-1.
% Thus, "impin" must be truncated to length corresponding to the final impulse.
% cc=0;      % initialize flag
%for qq=1:length(tiempo);
    %if tiempo(qq)>max(ti) & cc==0
    %cc=qq;      %note index corresponding to maximum time value
    %end
%end

```

```

%impin=impin(1:cc);

```

```

%shpcmd=conv(impin,command);
%shpcmd=shpcmd(1:length(command));
% plot the results

```

```

%orient tall
%subplot(2,1,1), plot(tiempo,impin,'red')
%title('Shaper Pulse Train ')
%ylabel('Amplitude')
%xlabel('Time, s')
%subplot(2,1,2), plot(command,'blue',shpcmd,'green');
%title('Unshaped and Shaped Input Commands')
%ylabel('Magnitude')
%xlabel('Increment')
end

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%MCONV.m: CONVOLVES TWO SHAPER IMPULSE TRAINS

```

```

%convolution of one pulse train to another
% m-# of modes to be cancelled (number of trains)
% n(i) - # of impulses for mode i (or in each train), i=1,.....,m

```

```

% shapin.m must be run first to put imp, ti in the workspace

```

```

if exist('tempp')==1
clear tempp tempt n impf if It;
end

```

```

m=vibs; %vibs defined in shapin.m
for i=1:m; %vibs is also # of modes to be cancelled
    [leng n(i)]=size(ti(i,:));
end;

```



```

tempp(1,:)=imp(1,:); %initialize tempp, tempp is a intermediate variable
tempt(1,:)=ti(1,:); %initialize tempt,tempt is a intermediate variable

    sumn=1;
    for k=1:1:m-1;
        for j=1:1:prod(n(1:k))
            tempp(k+1,((j-1)*n(k)+1):j*n(k))=imp(k+1,:)*tempp(k,j);
            tempt(k+1,((j-1)*n(k)+1):j*n(k))=ti(k+1,:)+tempt(k,j);
        end
    end

    impf=tempp(k+1,:); %the last row of tempp is the convolved impulse train
    [tif,It]=sort(tempt(k+1,:)); %sort new time for the convolved pulse train

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%STEPGEN.M: GENERATES SHAPED STEP COMMANDS
%generate shaped step command to use in simulink models

% generate shaped step command, using vector "tif" and "impf" from mconv.m

if exist('time')==1;
    clear sum1 sumt time;
end
tif=round(100*tif)/100;
sum1=0.0;
mark1=0;
[row1, coll]=size(tif);
lenk=2000;

for k=1:1:lenk;
    time(k)=20*(k-1)/lenk;
    for j=1:1:coll;
        if time(k)-tif(j)==0;
            sum1=sum1+impf(It(j));
            mark1=mark1+1
        end;
    end;
    sumt(k)=sum1;
end;

%for k=1:1:100;
%time(lenk+k)=(20-time(lenk))*(k-1)/100;
%sumt(lenk+k)=1;
%end

```

```

time=time';
sumt=sumt';
fprintf('      mark1=%f\n',mark1)
if mark1~=length(tif)
    fprintf('Some time points are missing, length of tif = %i\n',length(tif))
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%SMOOTH5.M: GENERATE A SMOOTHED OPEN LOOP TORQUE
COMMAND
%generate a trajectory by a polynomial function
function [smooth]=smp5(ttt)
x0=0;
xf=1;
t0=0;
tf=5;
x=ttt; %-t0;

if ttt<=tf;
    a=(35*tf^3*t0^4*xf-21*tf^2*t0^5*xf+7*tf*t0^6*xf-t0^7*xf+tf^7*x0-
        7*tf^6*t0*x0+21*tf^5*t0^2*x0-35*tf^4*t0^3*x0)/(tf-t0)^7;
    b=(140*tf^3*t0^3*(x0-xf))/(tf-t0)^7;
    c=(210*tf^2*t0^2*(tf+t0)*(x0-xf))/(t0-tf)^7;
    d=(140*tf*t0*(tf^2+3*tf*t0+t0^2)*(x0-xf))/(tf-t0)^7;
    e=35*(tf^3+9*tf^2*t0+9*tf*t0^2+t0^3)*(x0-xf)/(t0-tf)^7;
    f=84*(tf^2+3*tf*t0+t0^2)*(xf-x0)/(t0-tf)^7;
    g=70*(tf+t0)*(x0-xf)/(t0-tf)^7;
    h=20*(xf-x0)/(t0-tf)^7;

    sm1=a+b*x+c*x^2+d*x^3+e*x^4+f*x^5+g*x^6+h*x^7;
    sm2=b+2*c*x+3*d*x^2+4*e*x^3+5*f*x^4+6*g*x^5+7*h*x^6;
    sm3=2*c+6*d*x+12*e*x^2+20*f*x^3+30*g*x^4+42*h*x^5;
    sm4=6*d+24*e*x+60*f*x^2+120*g*x^3+210*h*x^4;
    sm5=24*e+120*f*x+360*g*x^2+840*h*x^3;

else

sm1=xf;
end
smooth=[sm1]; %sm2 sm3 sm4 sm5];

```

E. ROBUSTNESS ANALYSIS CODE

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%BATCHS.M
```

```
% BATCH file for command Shaping  
% batchs.m: study the robustness of input shaper and PWPF to  
% modal frequency and damping
```

```
if exist('Vfactorw')==1  
    clear Vfactorz Vfactorw  
end
```

```
if exist('ontime')==1  
    clear ontime onnum fer comtime  
    for n=2:1:9  
        eval(['clear vibq' int2str(n-1) ]);  
    end  
end
```

```
tf=20; %total simulation time  
slew=10; %degree  
countz=input('How many damping variations do you want to use?');  
zrange=input('What damping multiple do you want to use?');
```

```
countw=input('How many frequency variations do you want to use?');  
wrange=input('What frequency multiple do you want to use?');
```

```
if wrange==1;  
    wuncert=input('What frequency uncertainty do you want to use (wuncert*wn) ?');  
end
```

```
%countw=2;  
%countz=2;  
%vibs=4;  
tcomtime=0; %total computation time  
if exist('ti')==1;  
    clear ti imp vibmods modnum;  
end
```

```
fprintf(['There are ',num2str(mods), ' modes.\n'])  
vibs=input('How many modes do you want to cancel?');  
for jjj=1:vibs;  
    modnum(jjj)=input(['No.' num2str(jjj) ' mode you want to cancel is?']);  
end
```

```

go=input('What is the command initiation time (seconds)?')

type=input('Choose ZVD or ZVDD ("D" or "DD"),'s');
if strcmp(type,'D')~=1 & strcmp(type,'DD')~=1
    type=input('Try that again. What type: D or DD?',s');
end

for ii=1:1:countz
    factorz=zrange*ii/countz;
    Vfactorz(ii)=factorz;
    fprintf(' ii is %f\n',ii);
    for jj=1:1:countw
        tic;
        factorw=wuncert*wrangle*jj/countw;
        Vfactorw(jj)=factorw;
        [imp,ti]=shapbf(lambda,zi,modnum,go,type,vibs,factorw,factorz);
        if vibs>1
            [impf,tif,It]=mconvf(imp,ti,vibs);
            if countz~=1 | countw~=1,
                clear imp ti;
            end
        else
            impf=imp;
            tif=ti;
            for kk=1:1:length(ti)
                It(kk)=kk;
            end
        end
        [time,sumt]=stgenf(impf,tif,It);
        if countz~=1 | countw~=1,
            clear impf tif It;
        end
        %figure(1);
        %plot(time,sumt);
        %hold on;

        rk45('shp2sim',tf,zeros(1,2*mods+2+2),[1e-5,1e-5,1e-3,0,0,2]);
        clear time sumt;
        [ontime(ii,jj),onnum(ii,jj)]=ton(time1,PWPFO); %calling ton() function
        [len,wid]=size(time1);
        theta=states(:,1);

        mark3=0;
        for kk=1:1:len;
            if (time1(kk)-tf*0.75)<0.01;

```

```

        mark3=kk;
    end;
end;

sum2=0;
for k=mark3:1:len;
    sum2=sum2+abs(theta(k)-slew*pi/180);
end;
fprintf('sum2 is %f\n',sum2)
%fer=sum2/(len-mark3+1);
fer(ii,jj)=sum2/(len-mark3+1);
fprintf('    Final Error is %f\n\n', fer(mm,nn))

%calculate the index for each modal vibration
%the index is simply the average of the absolute value of the
%each modal displacement
for n=2:1:9
    eval(['vibq' int2str(n-1) '(ii,jj)=mean(abs(states(:, ' int2str(n) ')));']);
end

%save the data into 'data##' matrix
%eval(['data' int2str(ii) int2str(jj) '=[time1 states pdout pwpfout command];']);
if countz~=1 | countw~=1,
    clear states PWPFO time1
end
comtime(ii,jj)=toc
tcomtime=comtime(ii,jj)+tcomtime
end
end

if countz~=1 | countw~=1,
    figure
    mesh(Vfactorz,Vfactorw,fer);
    title('Final Stage Error (Rigid Body)');
    ylabel('Error ratio in damping');
    xlabel('Error ratio in modal frequency')

    figure
    mesh(Vfactorz,Vfactorw, vibq1);
    title('Average Absolute Displacement of Modal 1');
    ylabel('Error ratio in damping');
    xlabel('Error ratio in modal frequency')

    figure
    mesh(Vfactorz,Vfactorw, vibq2);

```

```
title('Average Absolute Displacement of Modal 2');  
ylabel('Error ratio in damping');  
xlabel('Error ratio in modal frequency')
```

```
figure  
mesh(Vfactorz,Vfactorw, vibq3);  
title('Average Absolute Displacement of Modal 3');  
ylabel('Error ratio in damping');  
xlabel('Error ratio in modal frequency')
```

```
figure  
mesh(Vfactorz,Vfactorw, vibq4);  
title('Average Absolute Displacement of Modal 4');  
ylabel('Error ratio in damping');  
xlabel('Error ratio in modal frequency')
```

```
figure  
mesh(Vfactorz,Vfactorw, vibq5);  
title('Average Absolute Displacement of Modal 5');  
ylabel('Error ratio in damping');  
xlabel('Error ratio in modal frequency')
```

```
figure  
mesh(Vfactorz,Vfactorw, vibq6);  
title('Average Absolute Displacement of Modal 6');  
ylabel('Error ratio in damping');  
xlabel('Error ratio in modal frequency')
```

```
figure  
mesh(Vfactorz,Vfactorw, vibq7);  
title('Average Absolute Displacement of Modal 7');  
ylabel('Error ratio in damping');  
xlabel('Error ratio in modal frequency')
```

```
figure  
mesh(Vfactorz,Vfactorw, vibq8);  
title('Average Absolute Displacement of Modal 8');  
ylabel('Error ratio in damping');  
xlabel('Error ratio in modal frequency')
```

```
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%IMPGENF.M: GENERATES & PLOTS A SHAPER IMPULSE SEQUENCE
```

```
% generate impule sequence based on impf and tif
% function [times,sumts]=impgenf(impf,tif,It)
```

```
function [times,out]=impgenf(impf,tif,It)
```

```
% generate impulse sequence, using vector "tif" and "impf" from mconv.m
```

```
%if exist('time')==1;
% clear sum1 sumt time;
%end
```

```
tif=round(100*tif)/100;
mark1=0;
[row1, coll]=size(tif);
lenk=2000;
```

```
for k=1:1:lenk;
    time(k)=20*(k-1)/lenk;
    out(k)=0;
    for j=1:1:coll;
        if time(k)-tif(j)==0;
            out(k)=impf(It(j));
            mark1=mark1+1;
        end;
    end;
```

```
end;
```

```
%for k=1:1:100;
%time(lenk+k)=(20-time(lenk))*(k-1)/100;
%sumt(lenk+k)=1;
%end
```

```
times=time';
out=out';
```

```
fprintf('      mark1=%f\n',mark1)
if mark1~=length(tif)
    fprintf('Some time points are missing, length of tif = %i\n',length(tif))
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%MCONVF.m: CALLABLE FUNCTION VARIANT OF MCONV
```

```
% function version of mconv
```

```
function [impf,tif,It]=mconvf(imp,ti,vibs)
```

```
%convolution of one pulse train to another
```

```
% m - # of modes to be cancelled (number of trains) (same as vibs)
```

```
% n(i) - # of impulses for mode i (or in each train), i=1,.....,m
```

```
% shapin.m must be run first to put imp, ti in the workspace
```

```
%if exist('tempp')==1
```

```
%clear tempp tempt n impf if It;
```

```
%end
```

```
m=vibs;
```

```
%[m,widthimp]=size(imp); %to get vibs
```

```
for i=1:m;
```

```
    [leng n(i)]=size(ti(i,:));
```

```
end;
```

```
tempp(1,:)=imp(1,:); %initialize tempp, tempp is a intermediate variable
```

```
tempt(1,:)=ti(1,:); %initialize tempt,tempt is a intermediate variable
```

```
    sumn=1;
```

```
for k=1:1:m-1;
```

```
    for j=1:1:prod(n(1:k))
```

```
        tempp(k+1,((j-1)*n(k)+1):j*n(k))=imp(k+1,:)*tempp(k,j);
```

```
        tempt(k+1,((j-1)*n(k)+1):j*n(k))=ti(k+1,:)+tempt(k,j);
```

```
    end
```

```
end
```

```
impf=tempp(k+1,:); %the last row of tempp is the convolved impulse train
```

```
[tif,It]=sort(tempt(k+1,:)); %sort new time for the convolved pulse train
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%STEPGENF.m: CALLABLE FUNCTION VARIANT OF STEPGEN.M
```

```
% function version of stepgen.m
```

```
function [times,sumts]=stgenf(impf,tif,It)
```

```
% generate shaped step command
```

```
% generate shaped step command, using vector "tif" and "impf" from mconv.m
```



```

%if exist('time')==1;
% clear sum1 sumt time;
%end

tif=round(100*tif)/100;

sum1=0.0;
mark1=0;

[row1, coll]=size(tif);

lenk=2000;
for k=1:1:lenk;
time(k)=20*(k-1)/lenk;
for j=1:1:coll;
    if time(k)-tif(j)==0;
        sum1=sum1+impf(It(j));
        mark1=mark1+1;
    end;
end;
sumt(k)=sum1;
end;

%for k=1:1:100;
%time(lenk+k)=(20-time(lenk))*(k-1)/100;
%sumt(lenk+k)=1;
%end

times=time';
sumts=sumt';

fprintf('    mark1=%f\n',mark1)
if mark1~=length(tif)
    fprintf('Some time points are missing, length of tif = %i\n',length(tif))
end

```

LIST OF REFERENCES

- Agrawal, B. N., Class Notes, AA4816 "Dynamics and Control of Smart Structures," Naval Postgraduate School, Monterey, California, 1996.
- Anthony, T., and Wie, B., "Pulse-Modulated Control Synthesis for a Flexible Spacecraft", *J. Guidance*, vol. 13, no. 6, November-December 1990.
- Crain, E., Singhose, W., Seering, W., "Derivation and Properties of Convolved and Simultaneous Two-Mode Input Shapers," 13th Triennial IFAC World Congress, San Francisco, CA, 1996.
- DeBra, D. "Pulse Modulators", AA277 Class Notes, Department of Aeronautics & Astronautics, Stanford University, 1980.
- Hailey, J. A., *Experimental Verification of Attitude Control Techniques for Flexible Spacecraft Maneuvers*, Master's Thesis, Naval Postgraduate School, Monterey, CA, March 1992.
- Hsu, H., *Signals and Systems Theory and Problems*, Schaum's Outline Series, McGraw-Hill, Inc. 1995.
- McClelland, R.S., *Spacecraft Attitude Control System Performance Using Pulse-Width, Pulse-Frequency Modulated Thrusters*, Master's Thesis, Naval Postgraduate School, Monterey, CA, 1994.
- Meyer, J.L. Harrington, W.B., Agrawal, B.N., and Song, G., "Application of Piezoceramics to Vibration Suppression of a Spacecraft Flexible Appendage," AIAA Guidance, Navigation, and Control Conference, San Diego, CA, 1996
- Meyer, J.L., Class Notes, AA4900 "Thruster Control of Flexible Spacecraft", Naval Postgraduate School, Monterey, CA, 1995.
- Pao, L. & Singhose, W., "A Comparison of Constant and Variable Amplitude Command Shaping Techniques for Vibration Reduction," 4th IEEE Conference on Control Applications, Albany, NY, 1995.
- Pao, L. & Singhose, W., "On the Equivalence of Minimum Time Input Shaping with Traditional Time-Optimal Control," 4th IEEE Conference on Control Applications, Albany, NY, 1995.
- Pao, L. "An Analysis of the Total Insensitivity of Input Shaping Designs," AIAA Guidance, Navigation and Control Conference, San Diego, CA 1996.

- Singer, N. & Seering, W., "Preshaping Command Inputs to Reduce System Vibration," *Transactions of the ASME*, Vol. 112, March 1990.
- Singer, N. *Residual Vibration Reduction in Computer Controlled Machines*, Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, MA, 1988.
- Singhose, W. & Pao, L., "Comparison of Input Shaping Techniques for Speed-Critical Multi-Mode Flexible Systems," *Proceedings of the 1996 IEEE International Conference on Control Applications*, Dearborn, MI, 1996.
- Singhose, W., Pao, L. & Seering, W., "Time-Optimal Rest-to-Rest Slewing of Multi-Mode Flexible Spacecraft Using ZVD Robustness Constraints," AIAA Guidance, Navigation, and Control Conference, San Diego, CA 1996.
- Smith, O.J.M., *Feedback Control Systems*, McGraw-Hill, New York, 1958, p.338.
- Watkins Jr., R.J., *The Attitude Control of Flexible Structures*, Master's Thesis, Naval Postgraduate School, Monterey, CA, June, 1991.
- Wie, B. and Plescia, C. "Attitude Stabilization of Flexible Spacecraft During Stationkeeping Maneuvers", *J. Guidance*, Vol. 7, No. 4, July-August 1984.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
8725 John J. Kingman Road, Suite 0944
Ft. Belvoir, VA 22060-6218 2
2. Dudley Knox Library
Naval Postgraduate School
411 Dyer Road
Monterey, CA 93943-5002 2
3. Chairman, Code AA
Department of Aeronautics and Astronautics
Naval Postgraduate School
Monterey, CA 93943 1
4. Professor Brij N. Agrawal, Code AA/Ag
Department of Aeronautics and Astronautics
Naval Postgraduate School
Monterey, CA 93943 2
5. Dr. Gangbing Song, Code AA/Gs
Department of Aeronautics and Astronautics
Naval Postgraduate School
Monterey, CA 93943 2
6. Professor I. Mike Ross, Code AA/Ro
Department of Aeronautics and Astronautics
Naval Postgraduate School
Monterey, CA 93943 1
7. LCDR Nicholas Buck, USN
7909 Foxhound Road
McLean, VA 22102 3

INITIAL DISTRIBUTION LIST

- | | | |
|----|--|---|
| 1. | Defense Technical Information Center 8725 John J. Kingman Road, Suite 0944 Ft. Belvoir, VA 22060-6218 | 2 |
| 2. | Dudley Knox Library Naval Postgraduate School 411 Dyer Road Monterey, CA 93943-5002 | 2 |
| 3. | Chairman, Code AA Department of Aeronautics and Astronautics Naval Postgraduate School Monterey, CA 93943 | 1 |
| 4. | Professor Brij N. Agrawal, Code AA/Ag Department of Aeronautics and Astronautics Naval Postgraduate School Monterey, CA 93943 | 2 |
| 5. | Dr. Gangbing Song, Code AA/Gs Department of Aeronautics and Astronautics Naval Postgraduate School Monterey, CA 93943 | 2 |
| 6. | Professor I. Mike Ross, Code AA/Ro Department of Aeronautics and Astronautics Naval Postgraduate School Monterey, CA 93943 | 1 |
| 7. | LCDR Nicholas Buck, USN 7909 Foxhound Road McLean, VA 22102 | 3 |